



MeMAD

Methods for Managing
Audiovisual Data

memad.eu
info@memad.eu

Twitter – @memadproject
LinkedIn – MeMAD Project

MeMAD Deliverable

D3.3 TV moments detection and linking (final version)

Grant agreement number	780069
Action acronym	MeMAD
Action title	Methods for Managing Audiovisual Data: Combining Automatic Efficiency with Human Accuracy
Funding scheme	H2020–ICT–2016–2017/H2020–ICT–2017–1
Version date of the Annex I against which the assessment will be made	23.6.2020
Start date of the project	1.1.2018
Due date of the deliverable	31.03.2021
Actual date of submission	27.04.2021
Lead beneficiary for the deliverable	EURECOM
Dissemination level of the deliverable	Public

Action coordinator's scientific representative

Prof. Mikko Kurimo

AALTO–KORKEAKOULUSÄÄTIÖ, Aalto University School of Electrical Engineering,
Department of Signal Processing and Acoustics
mikko.kurimo@aalto.fi



MeMAD project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780069. This document has been produced by the MeMAD project. The content in this document represents the views of the authors, and the European Commission has no liability in respect of the content.

Authors in alphabetical order		
Name	Beneficiary	e-mail
Thibault Ehrhart	EURECOM	thibault.ehrhart@eurecom.fr
Ismail Harrando	EURECOM	ismail.harrando@eurecom.fr
Ilkka Koskenniemi	LINGSOFT	ilkka.koskenniemi@lingsoft.fi
Mikko Kurimo	AALTO	mikko.kurimo@aalto.fi
Jorma Laaksonen	AALTO	jorma.laaksonen@aalto.fi
Tiina Lindh-Knuutila	LLS	tiina.lindh-knuutila@lingsoft.fi
Pasquale Lisena	EURECOM	pasquale.lisena@eurecom.fr
Dejan Porjazovski	AALTO	dejan.porjazovski@aalto.fi
Alison Reboud	EURECOM	alison.reboud@eurecom.fr
Raphaël Troncy	EURECOM	raphael.troncy@eurecom.fr

Internal reviewers in alphabetical order		
Name	Beneficiary	e-mail
Lauri Saarikoski	Yle	lauri.saarikoski@yle.fi
Karel Braeckman	Limecraft	karel.braeckman@limecraft.com

Abstract

This deliverable describes the final methods, tools and results developed within WP3. In particular, it extends the preliminary methods for detecting and enriching moments that were described in the Deliverable D3.2 and it completes and builds upon the MeMAD Knowledge Graph that has been initiated in the Deliverable D3.1.

This deliverable extends the methods for detecting so-called important moments in a video. The importance of a video sequence creating such a moment being highly subjective, we consider proxies such as memorability. In this reporting period, we have participated in the MediaEval 2020 Task on Predicting Media Memorability where we obtained the 3rd best results among the 7 participants. We also evaluated our approach on two MeMAD corpora in order to assess the generalization of our method when applied on very different video corpus in terms of genre and themes which is deceptive and shows that this is still an open research problem. One key aspect of the success of this task is to rely on an effective content segmentation. We tackled this problem by providing a new unsupervised method that we evaluate on the MeMAD corpora. We also address another common problem for video archives which consists in temporally aligning existing video content description with the media itself.

This deliverable extends also the methods for enriching important moments, mostly from textual inputs (e.g. ASR or descriptions provided by archivists) but also from the audio modality with an attempt to extract named entity directly from the speech. We introduce a new library and RESTful API named ToMoDAPI that enables to compare topic modeling algorithms. We provide a thorough and systematic comparison of existing methods in order to highlight the importance of numerous parameters depending on what objective has to be optimized. We also developed a new original method named ZeSTE that enables to predict the topics of a piece of content, while providing explanations leveraging on the ConceptNet common sense knowledge graph. Finally, we evaluate how we can categorize the MeMAD program sequences in terms of topics in a zero-shot setting, without relying on a pre-trained dataset. We have continued to provide new methods for extracting and disambiguating named entities, from ASR (textual input) or directly from the speech (audio input) and we have evaluated the results on the MeMAD corpora. We participated in the TRECVID 2020 Video Summarization task where we achieved the best performance by far.

Finally, we present MeMAD Explorer, a responsive web application that provides the functionalities of an exploratory search engine that is built on top of the MeMAD Knowledge Graph. We publish the MeMAD Knowledge Graph itself, enhanced by the numerous results coming from the multimodal media analysis performed in WP2 (ASR, face recognition, deep captioning), WP3 (Named Entity Recognition and Disambiguation, topics extraction, content segmentation) and WP4 (machine translation) obtained on medias from the MeMAD corpora.

This deliverable finally summarizes in an appendix the dissemination activities related to the research work in MeMAD's Work Package WP3 during its third year which amounts to 8 scientific publications (including one journal article) and 3 other submissions currently under review for this reported period.

Contents

1	Introduction	6
2	Moments Detection	7
2.1	Predicting Media Memorability in MediaEval 2020	7
2.1.1	Method	7
2.1.2	Results and Analysis	8
2.2	Predicting Media Memorability in MeMAD corpora	10
2.2.1	Yle Urheiluruutu	10
2.2.2	Surrey20	12
2.3	Unsupervised Multimodal Content Segmentation	16
2.3.1	Method	16
2.3.2	Textual Feature Extraction	17
2.3.3	Visual Feature Extraction	17
2.3.4	Modality Combination	17
2.3.5	Evaluation	18
2.4	Distant-Supervised Multimodal Content Segmentation	20
2.4.1	Method	20
2.4.2	Evaluation	21
3	Moments Enrichment	23
3.1	Extracting and Predicting Topics	23
3.1.1	TOMODAPI: A Topic Modeling API to Train, Use and Compare Topic Models	23
3.1.2	Watch Your Model: A Systematic Evaluation of Topic Models	23
3.1.3	Towards Zero-shot Explainable Topic Categorization Using a Common Sense Knowledge Graph	24
3.1.4	Zero-shot Theme Extraction on MeMAD corpora	24
3.2	Named Entity Recognition (NER) methodologies	26
3.2.1	GraphNER	26
3.2.2	Spoken NER	31
3.2.3	Evaluating Named Entity Linking (NEL) Approaches on MeMAD Data	36
3.3	Video Summarization	40
3.3.1	Approach	41
3.3.2	Results and Analysis	43
4	Exploring the MeMAD Knowledge Graph	45
4.1	MeMAD ontology and controlled vocabularies	45
4.2	MeMAD automatically generated metadata	46
4.3	MeMAD Exploratory Search Engine	50
5	Conclusion	52
6	References	54
A	Dissemination activities	58

B	Appendices	59
B.1	EURECOM and AALTO's MediaEval 2020 workshop paper	59
B.2	EURECOM and AALTO's TRECVID VSUM 2020 workshop paper	63
B.3	EURECOM's NLP-OSS 2020 workshop paper	67
B.4	EURECOM's ACL 2021 submission	77
B.5	EURECOM's ESWC 2021 poster paper	88
B.6	EURECOM's DataTV 2021a workshop paper	94
B.7	EURECOM's DataTV 2021b workshop paper	108
B.8	EURECOM's DHQ 2021 journal paper	115
B.9	EURECOM's LDK 2021 conference paper	144
B.10	EURECOM's SEMANTICS 2021 submission	160
B.11	Aalto's TSD 2021 submission	174

1 Introduction

Multimedia systems typically contain digital documents of mixed media types, which are indexed on the basis of strongly divergent metadata standards. This severely hampers the inter-operation of such systems. Therefore, machine understanding of metadata coming from different applications is a basic requirement for the inter-operation of distributed multimedia systems. Furthermore, the content will be processed by automatic multimedia analysis tools which have their own formats for exchanging their results. One of the main goals of MeMAD is to enrich seed video content with additional content that come from diverse sources including media archives and encyclopedia resources.

The general methodology that we follow consists of: i) semantifying the legacy metadata coming with audiovisual content (program metadata coming from the producer, the broadcaster and/or the archive) and ii) automatically extracting concepts and entities from the subtitles or the text generated by automatic speech recognition on the audiovisual content. The resulting knowledge graph can then be used to infer additional information in order to enrich and hyperlink key video content moments.

In this deliverable, we describe how we have consolidated the methods for detecting so-called important moments in a video. The importance of a video sequence creating such a moment being highly subjective, we consider proxies such as memorability. In this reporting period, we have participated in the MediaEval 2020 Task on Predicting Media Memorability where we obtained the 3rd best results among the 7 participants. We have also evaluated our approach on two MeMAD corpora in order to assess the generalization of our method when applied on very different video corpora in terms of genre and themes. One key aspect of the success of this task is to rely on an effective content segmentation. We tackled this problem by providing a new unsupervised method that we evaluate on the MeMAD corpora. We also address another common problem for video archives which consists in temporally aligning existing video content description with the media itself (Section 2).

We have developed new methods for enriching important moments. In particular, we introduce a new library and RESTful API named ToMoDAPI that enables to compare topic modeling algorithms. We provide a thorough and systematic comparison of existing methods in order to highlight the importance of numerous parameters depending on what objective has to be optimized. We develop a new original method that enables to predict the topics of a piece of content, while providing explanations leveraging on the ConceptNet common sense knowledge graph. Finally, we evaluate how we can categorize the MeMAD program sequences in terms of topics in a zero-shot setting, without relying on a pre-trained dataset. We have continued to provide new methods for extracting and disambiguating named entities, from ASR or directly from the speech and we evaluate the results on the MeMAD corpora. We participated in the TRECVID 2020 Video Summarization task where we achieved the best performance by far (Section 3).

Finally, we present MeMAD Explorer, a responsive web application that provides the functionalities of an exploratory search engine that is built on top of the MeMAD Knowledge Graph. We publish the MeMAD Knowledge Graph itself, enhanced by the numerous results coming from the multimodal media analysis performed in WP2 (ASR, face recognition, deep captioning), WP3 (Named Entity Recognition and Disambiguation, topics extraction, content segmentation) and WP4 (machine translation) obtained on media from the MeMAD corpora (Section 4).

We conclude this deliverable by providing future research directions (Section 5) and we list our dissemination activities (Section A). The Appendixes contain the 8 accepted papers as well as the 3 submissions which are still under review (Section B).

2 Moments Detection

2.1 Predicting Media Memorability in MediaEval 2020

Considering video memorability as a useful tool for digital content retrieval as well as for sorting and recommending an ever growing number of videos, the MediaEval Predicting Media Memorability task aims at fostering the research in the field by asking its participants to automatically predict both a short and a long term memorability score for a given set of annotated videos. The full description for this task is provided in [1].

We described in Deliverable D3.2 [2] our 2019 approach for tackling this challenge. We obtained the best score for the long term memorability prediction among all participants [3] while [4] obtained the best score on short term memorability prediction. Both methods rely on multimodal features. More precisely, we used the textual and visual modalities to provide predictions and to operate a weighted average to obtain the final scores. We also experimented with a visiolinguistic representation as an alternative to a weighted average that we presented in the revised version of Deliverable D3.2 [2].

We emphasised on two critical limitations with the MediaEval 2019 dataset. First, the audio was muted (there was no sound or speech). Second, the videos were very short and did not contain a lot of actions. The videos provided in the 2020 edition overcame some of these limitations: they contained more actions and included sound (despite generally not containing any speech). Consequently, we adapted our approach and included audio features as well as video features (rather than middle frame image features only) to our model. Finally, a key contribution of our approach is to show that visiolinguistic representations also had complementary information to the text and vision scores. Our final model is a multimodal weighted average with visual and audio deep features extracted from the videos, textual features from the provided captions and visiolinguistic features. We published our method which is open sourced at <https://github.com/MeMAD-project/media-memorability>.

2.1.1 Method

We trained separate models for the short and long term predictions using a 6-fold cross-validation of the training set, which means that, given a total of 590 videos, we used 492 videos for training and 98 videos for testing each model. This setup was useful to evaluate our model. Once the best parameters were known, we generated a new model trained on the 590 videos that was used on the test set that the sole organizers can evaluate.

Audio-Visual Features Our audio-visual memorability prediction scores are based on using a feed-forward neural network with a concatenation of video and audio features in the input, one hidden layer of units and one unit in the output layer. The best performance was obtained with 2575-dimensional features consisting of the concatenation of 2048-dimensional I3D [5] video features and 527-dimensional audio features. Our audio features encode the occurrence probabilities of the 527 classes of the Google AudioSet Ontology [6] in each video clip. The hidden layer uses ReLU activation¹ and dropout during the training phase, while the output unit is sigmoidal. The training of the network used the Adam optimizer [7]. The features, the number of training epochs and the number of units in the hidden layer were selected with the 6-fold cross-validation. For short term memorability prediction, the optimal number of epochs was 750 and the optimal hidden layer size 80 units, whereas for the long term prediction these figures were 260 and 160, respectively.

We also experimented with other types of features and their combinations. These include the ResNet [8] features extracted just from the middle frames of the clips as this approach

¹[https://en.wikipedia.org/wiki/Rectifier_\(neural_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

worked very well in 2019. The contents of the videos provided in 2020 are, however, such that genuine video features I3D and C3D [9] work better than still image features. When I3D and AudioSet features are used, C3D features do not bring any additional advantage.

Textual Feature We leverage the video descriptions provided by the organizers. First, all the provided descriptions are concatenated by video identifier to get one string per video. To generate the textual representation of the video content, we used the following methods:

- Computing TF-IDF, removing rare (less than 4 occurrences) and stopwords and accounting for frequent 2-grams.
- Averaging GloVe embeddings for all non-stopwords words using the pre-trained 300d version [10].
- Averaging BERT [11] token representations (keeping all the words in the descriptions up to 250 words per sentence).
- Using Sentence-BERT [12] sentence representations. We use the distilled version that is fine-tuned for the STS Textual Similarity Benchmark².

For each representation, we experimented with multiple regression models and fine-tuned the hyper-parameters for each model using the 6-fold cross-validation on the training set. For our submission, we used the *Averaging GloVe embeddings* with a Support Machine Regressor with an RBF kernel and a regulation parameter $C = 1e - 5$.

We also attempted enhancing the provided descriptions with additional captions automatically generated using the DeepCaption³ software developed by the partner Aalto University. We did not see an improvement in the results, which is probably due to the nature of the clips provided for this year’s edition (as DeepCaption is trained on static stock images from MS COCO and TGIF datasets).

Visiolinguistic Features ViLBERT [13] is a task-agnostic extension of BERT that aims to learn the associations and links between visual and linguistic properties of a concept. It has a two-stream architecture, first modelling each modality (i.e. visual and textual) separately, and then fusing them through a set of attention-based interactions (co-attention). ViLBERT is pre-trained using the Conceptual Captions data set (3.3M image-caption pairs) [14] on masked multi modal learning and multi-modal alignment prediction. We used a frozen pre-trained model which was fine-tuned twice, first on the task of Video-Question Answering (VQA) [15] and then on the 2019 MediaEval Memorability task and dataset.

The 1024-dimensional features extracted for the two modalities can be combined in different ways. In our experiment, multiplying textual and visual feature vectors performed the best for short term memorability prediction. However, using the sole visual feature vectors worked better for long term memorability prediction. Averaging the features extracted from 6 frames performed better than only using only the middle frame. We experimented with the same set of regression models as for the textual approach. In our submission, we used a Support Machine Regressor with a regulation parameter $C = 1e - 5$ and an RBF or Poly kernel respectively for short and long term scores prediction.

2.1.2 Results and Analysis

We prepared 5 different runs following the task description defined as follows:

- MeMAD1 = Audio-Visual Score

²<https://huggingface.co/sentence-transformers/distilbert-base-nli-stsb-mean-tokens>

³<https://github.com/aalto-cbir/DeepCaption>

- MeMAD2 = Visiolinguistic Score
- MeMAD3 = Textual Score
- MeMAD4 = $0.5 * \text{run1} + 0.2 * \text{run2} + 0.3 * \text{run3}$
- MeMAD5 = MeMAD4 with LT scores for LT task

For the Long Term task, all models except *run5* use exclusively short-term scores. For runs 4 and 5, we normalise the scores obtained from runs 1, 2 and 3 before combining them.

Table 1: Average Spearman score obtained on a 6-folds cross validation of the Training set

Method	Short Term	Long Term
MeMAD1	0.2899	0.179
MeMAD2	0.214	0.1309
MeMAD3	0.2506	0.1372
MeMAD4	0.3104	0.2038
MeMAD5	0.067	0.1700

Table 1 provides the Spearman score obtained for each run when performing a 6-folds cross-validation on the training set. We observe that our models use only the training set, as the annotations on the later-provided development set did not yield better results. We hypothesize that this is due to the fewer number of annotations per video available as many videos had a score for 1 which we do not observe on the training set.

We present in Table 2 the final results obtained on the test set using models trained on the full training set composed of 590 videos. We observe that the weighted average method which uses short term scores works the best for both short and long term prediction, obtaining results which are approximately double the mean Spearman score obtained across all the other teams. This approach also ranks second for the long term memorability and third for the short term memorability.

Table 2: Results on the Test set for Short Term (ST) and Long Term (LT) memorability

Method	SpearmanST	SpearmanL
MeMAD1	0.099	0.077
MeMAD2	0.098	-0.017
MeMAD3	0.073	0.019
MeMAD4	0.101	0.078
MeMAD5	0.101	0.067
AvgTeams	0.058	0.036
DCU-Audio	0.137	0.113
MG-UCB	0.136	0.077
CUC-DMT	0.06	0.049
KT-UPB	0.053	0.037
Essex-NLIP	0.042	0.043
DCU@ML-Labs	0.034	-0.01
GTHU-UPM	0.016	-0.041
MMSys	0.007	0.048

The other best teams also proposed multimodal models. In particular, the team who ranked first in both tasks investigated the relevance of audio gestalt to evaluate the weight of the audio modality on overall video memorability [16]. The team ranking second for short term

memorability [17] showed that the audio modality performs relatively well for predicting short-term memorability.

Our best results on the test set are however significantly worse than the ones we obtained on average over the 6-folds of the training set suggesting that the test set is quite different from the training set. The results for Long Term prediction are always worse than the ones for Short Term prediction. Finally, even the best team scores are below the ones obtained for the 2018 and 2019 videos showing the difficulty of this task and the challenges ahead for their generalization across video genres, duration and themes.

We have described a multimodal weighted average method proposed for the MediaEval 2020 Task on Predicting Media Memorability. One of our key contributions is to have empirically demonstrated that video features performed the best in comparison to image, audio and text separately. Similarly to last year, short term scores predictions correlated better with long term scores than the predictions made when training directly on long term scores. Finally considering the difference of results obtained between the training and test set, it would be interesting to investigate further the differences between these datasets in terms of content (video, audio and text) and annotations. We conclude that generalizing this type of task to different video genres and characteristics remains a scientific challenge. The 2020 dataset is closed to real-world in-domain data (if we compare with the 2019 dataset) but these academic benchmarks still lack longer video content and detailed annotations.

2.2 Predicting Media Memorability in MeMAD corpora

Despite the limitations described above in terms of generalization of the method for predicting memorable moments, we have attempted to predict the memorability scores of segments from MeMAD videos using the ensemble approach we developed for MediaEval 2020. Our goal is to assess the robustness of our approach when being confronted with a very different dataset. We do not have ground truth annotations corresponding to short-term or long-term memorability scores that could be used for training or even for testing. We have envisioned to leverage on viewer data for programs available via IPTV, since fine-grained analytics is potentially available, considering that viewing peaks would match interesting moments. However, this data was mostly flat without highlighting clear peaks.

We selected two MeMAD datasets:

- Yle Urheiluruutu: 12 episodes of a sport magazine program⁴
- Surrey20: 20 movie excerpts with a narrative arc⁵

As we do not have any ground truth available for these datasets, we perform a post-hoc qualitative analysis, mainly focusing on the 6 segments predicted to be the most and the least memorable moments. We only considered short term memorability, since in both our 2019 and 2020 approaches, we found out that predicted short term memorability was a better proxy for the true long term score than the predicted long term score.

2.2.1 Yle Urheiluruutu

Urheiluruutu is a Finnish sports magazine program by Yle highlighting the sports events and results of that day. Each episode is very short (3-5 minutes) but there are also longer programs weekly (up to 20 minutes) in which some sports phenomena are discussed in more detail. The Yle Urheiluruutu dataset is composed of 12 episodes published every day between January 6,

⁴Production 3380 in Flow at <https://platform.limecraft.com/memad/#productions/3380/material/>

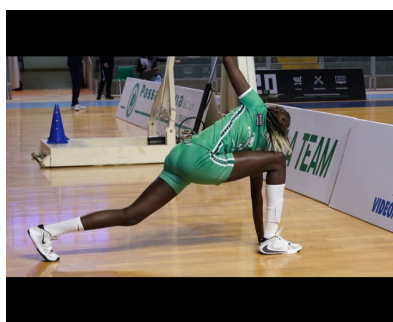
⁵Production 4236 in Flow at <https://platform.limecraft.com/memad/#productions/4236/material/>

2021 and January 17, 2021. Each episode lasts from 4 to 20 minutes, the ones being published on Saturday and Sunday being at least twice longer than the ones published on week days.

We chose shots as our segment unit and we computed the memorability score per shot. As opposed to the MediaEval task setting, we did not have human written captions for every segment. For the textual part, we therefore solely rely on automatically generated deep captions using the PicSOM tool developed as part of WP2. Figure 1 and 2 show respectively the middle frame and the deep captions of some of the **most** and **least** memorable Urheiluruutu segments. We specifically chose shots from different videos (among the 12 programs) and different parts (of a program).



(a) a man in a hat is standing in the water



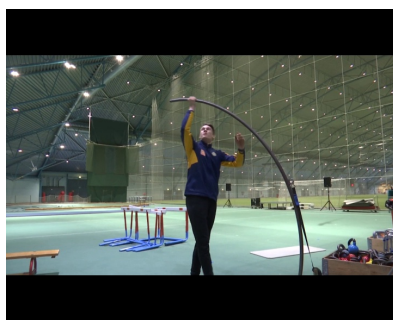
(b) a woman is jumping in the air on a court



(c) a man is playing a guitar while wearing a hat



(d) tennis player is holding a racket on a court



(e) a man holding a tennis racquet on a tennis court



(f) a woman is dancing and singing on stage

Figure 1: Middle frame and deep caption of some of the **most** memorable Urheiluruutu segments

We observe that the middle frame of the most memorable moments are much more diverse visually than the images from the least memorable moments, which are almost all ice hockey scenes. We also see a couple of faces in the most memorable segments and none in the least memorable ones. This is in line with the 2019 MediaEval dataset where a lot of video with faces were considered memorable.

In terms of automatically generated deep captions, we observed that the sports are often misidentified for the most memorable segments. In particular, 2 captions out of those 6 examples wrongly mention the sport 'tennis'. However for the least memorable segments, 'hockey' was once correctly identified. For the four other hockey pictures, the captions mention 'ski' which is incorrect but related to winter sport nevertheless. Based on these observations, we performed a keyword search in the deep captions of the whole dataset. It showed that the words 'hockey' and 'ski' become more frequent as the memorability score of the captions drops. The keyword 'tennis', on the contrary, is more frequent in the top memorable captions.



(a) a group of people on skis in the snow



(b) a hockey player is unk a goal



(c) a man riding skis down a snow covered slope



(d) a man is running and then unk his arms



(e) a man is dancing and singing in a room



(f) a man riding skis down a snow covered slope

Figure 2: Middle frame and deep caption of some of the **least** memorable Urheiluruutu segments

2.2.2 Surrey20

The Surrey dataset is composed of 20 movies excerpts and it has been thoroughly described in Deliverable D5.2. For this dataset, we have considered two different initial segmentations in order to predict the memorability score of each segment. First, we consider a simple shot segmentation as we did for the Yle Urheiluruutu sport magazines (Section 2.2.1). Second, we consider the Story Grammar annotations which were human made as part of the Deliverable D5.2. We also rely on automatically generated deep captions. Figures 3 and 4 show respectively the middle frame and deep captions of some of the **most** and **least** memorable Surrey20 shots. For all Figures, each segment is from a different film excerpt.

Shot segmentation. From Figures 3 and 4, we observe that the generated deep captions seem to be more correct than the ones generated for the Urheiluruutu videos. However, it is difficult to observe any features from the captions or images that would be specific to the most or least memorable shots groups. For example in terms of captions, 'a man is sitting in a room and talking' ((b) in Figure 3) and 'a man is lying on a bed and talking' ((e) in Figure 4) seem pretty close but their memorability ranking is not.

An interesting example, however, is the image (b) from Figure 3 and image (d) from Figure 4 depicting different moments of the same scene. Our approach predicted that the image with the face of the person would be memorable whereas the one without his face is one of the least memorable segment.

Overall, the results we observe might suggest that predicting what is memorable in a movie is a task that requires considering other aspects such as dialogue or information about the story. That is why, we did an experiment with the same dataset but with longer segments created by an annotator who segmented the video where the story grammar was changing.



(a) a man is driving a car and looking at something



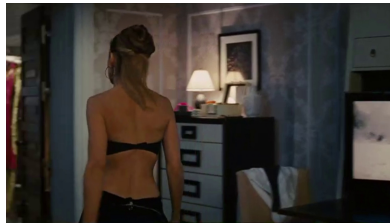
(b) a man is sitting in a room and talking



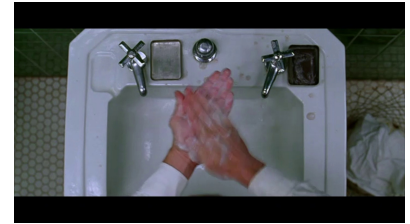
(c) a woman is walking down a hallway with a man



(d) a woman is laying on her stomach and smiling



(e) a woman is dancing in a room

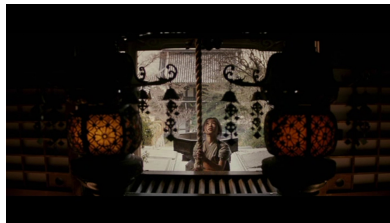


(f) a person holding a camera in a bathroom

Figure 3: Middle frame and deep caption of some of the **most** memorable Surrey shots



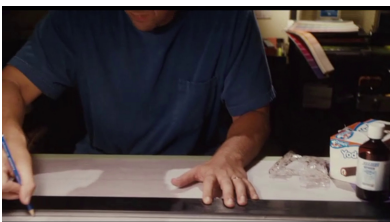
(a) a man is walking through a door and then he stops



(b) a man is jumping on a chair and then falls



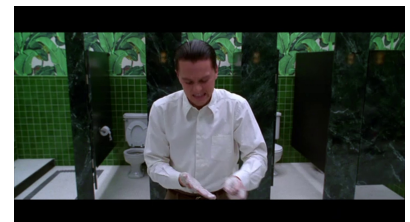
(c) a man is putting his hand on his face



(d) a man is typing on a computer keyboard

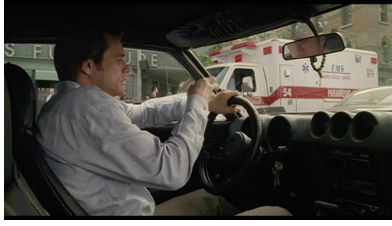


(e) a man is lying on a bed and talking



(f) a man in a suit and tie is standing in front of a window

Figure 4: Middle frame and deep caption of some of the **least** memorable Surrey shots



(a) DeepCaption: A man in a car with a cell phone.

(b) Subtitle: Bruce: Yep, yep. Meeting started. Without me. This is my luck. This is my luck!

(c) Story Grammar: Consequence



(j) DeepCaption: a man is lying on the ground and junk his head

(k) Subtitle: Julio: Wake up dad! Dad wake up!

(l) Story Grammar: Consequence



(d) DeepCaption: a woman is walking down a path with a child

(e) Subtitle: No dialogue in this sequence

(f) Story Grammar: Internal Response



(m) DeepCaption: a man is sitting in a chair and smiling

(n) Subtitle: Andy: Happy birthday.

(o) Story Grammar: Plan



(g) DeepCaption: a woman is walking through a door and then falls

(h) Subtitle: Carrie: It's a little loud

(i) Story Grammar: Initiating Event



(p) DeepCaption: a woman is smiling and looking down

(q) Subtitle: Jenny: Heh heh heh. David: No? Alright, up to you

(r) Story Grammar: Reaction

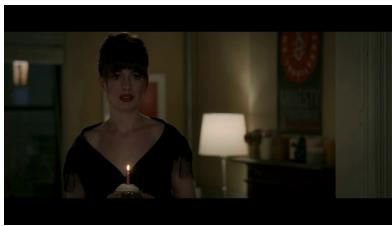
Figure 5: Some of the **most** memorable Surrey Story Units segments

Story Grammar segmentation. For this experiment, we made use of the dialogue that we concatenated with the automatically generated deep captions. Figures 5 and 6 show the middle frame, deep captions, subtitles and Story Grammar label of respectively some of the most and least memorable Surrey20 segments.

We observe that the memorable segments are completely different from the ones selected using a simpler shot segmentation. Among the most memorable segments, there is: one woman smiling, an injured man in the ground and someone with a birthday cake. These events are diverse but seem to be accurate candidates for memorable moments. We also can see that a large majority of the least memorable segments do not contain dialogues or only very short ones. This is interesting because our model was not trained on any dialogue data, but rather on captions. These results suggest that our model was able to somehow integrate the dialogue information. If we have a closer look at the subtitles of the most memorable segments, we find 'happy birthday', 'Wake up dad! Dad wake up!', or 'This is my luck. This is my luck!', which as far as a human can tell, seem to be potentially memorable moments.

Each of the segment is associated to a Story Segment which was not used as an input to our model. We can see that in the least memorable segments, four out of six are labeled as 'Setting' when none of the most memorable moments have this label. The most memorable moments, on the contrary, do not seem to be associated with one Story Segment in particular.

In conclusion, we have experimented with two different genres of videos (sport magazines and excerpt of movies), as well as two different types of segmentation (shot segmentation and human generated Story Grammar segments). It is not possible to compare these results with the ones obtained on MediaEval due to the lack of a ground truth. However, these experiments showed some interesting observations that would need to be further researched. First,



(a) DeepCaption:

(b) Subtitle: Nate: You look really pretty.

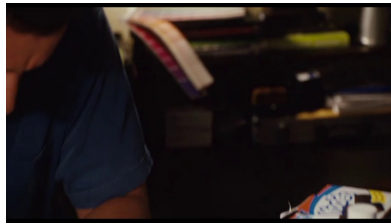
(c) Story Segment: Consequence



(j) DeepCaption: a man is looking at something and then looks away

(k) Subtitle: Julian: Uhh... RADIO: ...Sixth Avenue freeway is tied up around Lincoln, but six eighty-five is looking just dandy in both directions... more traffic reports on the 'Five' ... but coming up ...

(l) Story Grammar: Setting



(d) DeepCaption: a man is sitting in a chair and junk_g his head

(e) Subtitle: No dialogue in this sequence

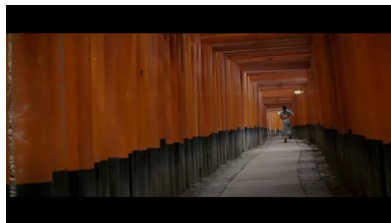
(f) Story Grammar: Setting



(g) DeepCaption: a man laying on a bed with a woman

(h) Subtitle: No dialogue in this sequence

(i) Story Grammar: Setting



(m) DeepCaption: a man is walking down a hallway and then falls

(n) Subtitle: No dialogue in this sequence

(o) Story Grammar: Setting



(p) DeepCaption: a man is looking at a woman and then she looks away

(q) Subtitle: No dialogue in this sequence

(r) Story Grammar: Reaction

Figure 6: Some of the **least** memorable Surrey Story Grammar segments

our model considers some sports to be more memorable than others (tennis versus ski/hockey) which may be correlated to their frequency. Second, for movies, we demonstrate that using different segmentations produce very different results. The Story Grammar segmentation’s results are more easily interpretable for humans. This could suggest that an adequate segmentation is an important requirement to obtain meaningful results. These results also suggested that using subtitles might be useful to our model, despite not having been trained on it. Finally, the Story Grammar ‘setting’ is very represented in the least memorable segments. These preliminary conclusions have encouraged us to work further on providing automatic methods for segmenting the content that we describe in the next section.

2.3 Unsupervised Multimodal Content Segmentation

Content segmentation is the process of dividing a document into smaller coherent units of meaning. In our use-case which is also described in Deliverable D7.4, we aim to segment long-duration programs into smaller segments: each treating a different topic or subject matter. This process is crucial both for editing raw content for production as well as re-purposing it for end users who may only be interested in specific topics or themes. However, it is a very time-consuming task to perform manually as it requires an annotator to watch the entirety of the program, and to precisely decide where units of content begin and end, even when it is supported by a well-performing shot segmentation. We propose a multimodal approach to automatically segment audiovisual media based on deep neural representations of both its textual and visual content.

2.3.1 Method

We define the task of content segmentation as follows: given a video representing a full program (e.g. an episode of news broadcast) as input, our method produces *segment boundary candidates*, i.e. timestamps on the program runtime at which a topical segment is likely to end and for the next one to start.

To do so, we consider the program as a sequence of two modalities:

- **Visual:** every program is a sequence of *shots* (i.e. a series of frames that runs for an uninterrupted period of time from the same camera and the same angle). To produce these shots, we use Flow’s shot segmentation service as described in the Deliverable D2.1.
- **Textual:** we run every program through the Automatic Speech Recognition service to produce automatic subtitles for it. The program is thus represented as a sequence of sentences produced by the ASR.

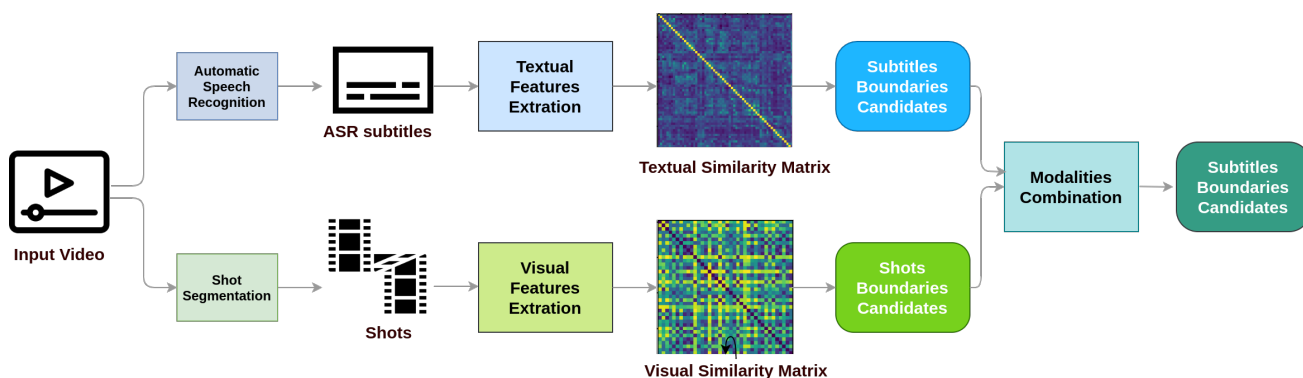


Figure 7: The unsupervised content segmentation pipeline (v1)

Figure 7 shows the main building blocks of the pipeline, the main components are further explained below.

2.3.2 Textual Feature Extraction

Once the automatic subtitles are generated, we transform every sentence into a fixed-dimensional vector, allowing us to compare the content throughout the program runtime. We use a pretrained multilingual Sentence-BERT [18] to encode all sentences from the subtitles into a common 300-d representation space. Following that, we compute the similarity matrix for all the textual content, i.e. we compute the cosine similarity between every sentence in the program. Under the hypothesis that two sentences that are close in the representation space have similar meaning and thus talk about the same topic, we generate candidates for the *segment boundaries* at the sentences where this similarity is lowest, which suggests that the topic has likely changed.

2.3.3 Visual Feature Extraction

Similarly to the textual similarity matrix, we form another matrix that describes the visual similarity between the visual shots of the program. The creation of the similarity matrix starts with the extraction of vectors of visual features that characterise the contents of each shot. In our approach, we have used the 2048-dimensional ResNet-152 feature extractor [8] and applied it to the middle frame of each shot. After extracting the features, we calculate pair-wise Euclidean distances between feature vectors, each representing one shot, and obtain a symmetric square matrix in which the diagonal values are zero and all off-diagonal values non-negative.

2.3.4 Modality Combination

In the next step, we combine the two matrices, similarity matrix of textual features and distance matrix of visual features, to form a multimodal distance matrix. The best predictions for the locations of topic or subject changes in the program can then be found in that matrix. This process consists of seven steps:

1. We resample each matrix with the temporal resolution of one second. As a result, both matrices get the same dimensionality which is larger than their original dimensions. As a visual shot and a spoken sentence usually lasts longer than one second, the resampling means in practice that the rows and columns of the matrices are duplicated or expanded to reflect the duration of the corresponding textual sentences and visual shots.
2. We multiply the values of the resampled textual similarity matrix with minus one to change the direction of the values to match that of the visual distance matrix.
3. We equalise the values of each matrix to the range $[0, 1]$. Equalisation is obtained by ordering the values of the matrix in the ascending order while storing mutually equal values in the same bins. After that, the indices of the ordering, divided by the number of elements in the matrix, form a value-equalised version of the distances scaled non-linearly to the range $[0, 1]$.
4. We create a combined multimodal distance matrix between pairs of seconds in the program by element-wise summing of the two unimodal matrices. As an alternative for summing, we also considered element-wise multiplication of the values.

5. We then re-equalise the values of the multimodal distance sum matrix to the range $[0, 1]$ in a manner similar to the process described in Step 3.
6. Next, we form a *segmentation score* that quantifies how different the contents of each particular second of the program is from its preceding and succeeding seconds. For this purpose, we sum the differences of the distance matrix values in a sliding time window that expands to w seconds in the past and the same number of seconds in the future. More precisely, we define the segmentation score $s(t)$ at time t as

$$s(t) = \sum_{i=1}^w (m[t, t-i] - m[t-1, t-i]) + \sum_{i=0}^{w-1} (m[t-1, t+i] - m[t, t+i]), \quad (1)$$

where $m[i, j]$ are values of the equalised multimodal distance matrix and the terms involving matrix values that are not defined are ignored. This segmentation score is then normalised by dividing each value $s(t)$ with the maximum of all $s(t)$. In our experiments we used the window length $w = 30$ seconds.

7. Finally, we select the predicted topic change positions by locating the maximum value of $s(t)$. After selecting the maximum position t^* and storing the corresponding segmentation score value $s(t^*)$, we apply *non-maximum suppression* and set the located maximum segmentation value and all preceding and succeeding values less than l seconds apart from it to zero. This step is repeated as long as positive segmentation score values can be found. In our experiments we used different l values for different content types.

Figure 8 shows an example how the textual similarity matrix and the visual distance matrix are combined in unsupervised multimodal content segmentation. The top-left matrix shows the similarities between the 61 textual sentences in the program. In that plot, the bright colours correspond to large similarity values, whereas in the remaining plots the dark shades correspond to small distance values. The top-center matrix is the textual distance matrix after resampling the original matrix to the resolution of one second, changing the direction of the values, and equalising the values to the range $[0, 1]$, as described in Steps 1–3 in the above process description. Similarly, the bottom-left matrix shows the original visual feature distance matrix between the 38 visual shots of the program and the bottom-center matrix displays the same matrix after second-based resampling and value equalisation. The bottom-right matrix is the result of Steps 3–4, where the two unimodal matrices have been summed and re-equalised. The top-right matrix shows the alternative version obtained with multiplication. The vertical lines in the bottom panes below the matrices show the $[0, 1]$ -normalised values of the segmentation score defined in Step 6. Its maximum locations, after non-maximum suppression described in Step 7, are then used as the predicted topic change positions.

2.3.5 Evaluation

To evaluate our approach, we perform a quantitative analysis using metrics from the literature, as well as a qualitative analysis of the output on some episodes for which we have both the segmentation ground truth (provided as metadata) and the automatically generated subtitles. While the method is unsupervised (it does not have a training component), all the used models are pretrained and the evaluation requires manually-annotated content to compare our results to human judgement.

For the automatic evaluation, we use the two standard metrics known as *WindowDiff* [19] and P_k [20], defined as follows:

- P_k : is the probability that two sentences drawn randomly from the program are correctly identified as belonging to the same segment or not belonging to the same segment. The

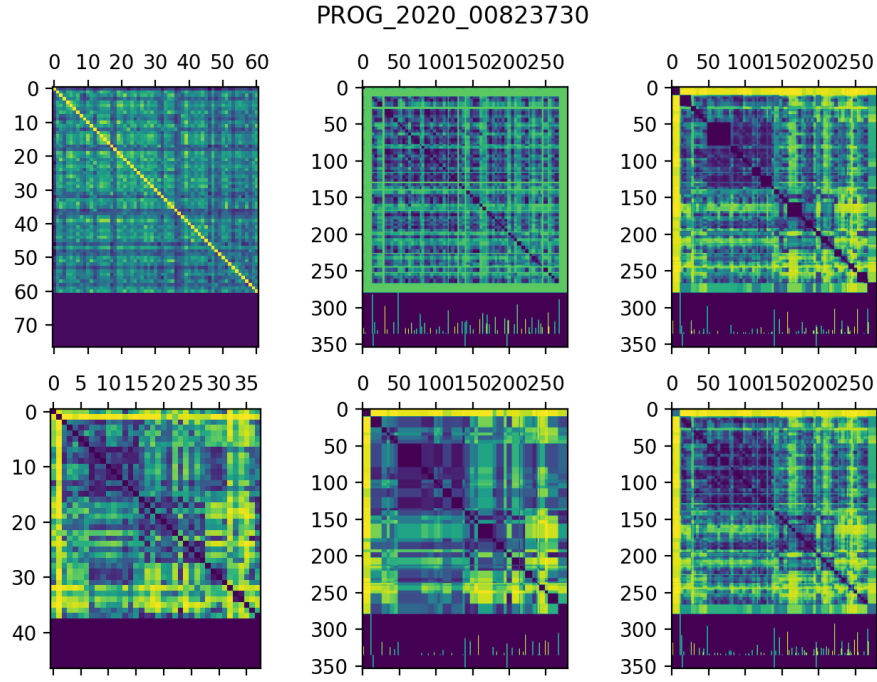


Figure 8: Stages of modality combination for unsupervised multimodal content segmentation. See text for the processing steps and the description of the matrices.

metric assigns penalties via a moving window of length k . At each location, it determines whether the two ends of the probe are in the same or different segments in the reference segmentation, and increases a counter if the algorithm’s segmentation disagrees.

- *WindowDiff*: a variant of the P_k measure, which penalizes false positives and near misses equally[21]. WindowDiff uses a sliding window over the segmentation; each window is evaluated as correct or incorrect. WindowDiff is effectively $1 - accuracy$ for all windows, but accuracy is sensitive to the balance of positive and negative data being evaluated.

For both metrics, an algorithm that assigns all boundaries correctly receives a score of 0 (the lower these scores are, the better the segmentation is). Both metrics use a tolerance parameter, k , which is recommended to be set to half of the average true segment size across the evaluated corpus.

Datasets. We conduct our evaluation protocol on the following datasets.

- **INA-44:** We select a subset of INA programs for which we have a manually-made segmentation for the content of the program. We only select programs that have been mostly segmented (dropping the ones where the segmentation covers less than 60% of the content of the program). This leaves us with 44 programs with varying properties (duration, number of segments, topics, etc).
- **Yle Urheiluruutu:** We select 12 episodes from *Urheiluruutu*, a sports news program, for which we have a manually-made segmentation. The episodes vary a little in characteristics, but they are all presented in the same format.

Quantitative Evaluation. We report the automatic evaluation results in Table 3. We remind that the tolerance parameter k is computed as follows:

$$k = \text{ceil}(\frac{1}{2} * \frac{\text{Average number of sentences per program}}{\text{Average number of segments per program}})$$

Dataset	N. videos	Avg. N. Segments	Avg. N. sentences	k	P_k	<i>WindowDiff</i>
INA-44	44	15	171	6	0.39	0.54
Yle Urheiluruutu	11	5	155	20	0.36	0.44

Table 3: Automatic segmentation results on subsets of the MeMAD dataset

The quantitative analysis shows roughly that about 39% and 36% of sentences, in INA-44 and Yle Urheiluruutu respectively, fall outside of the segment they were supposed to be (taken into account the tolerance parameter k), which could be in part due to the errors in the automatically generated subtitles. The higher scores of *WindowDiff* show that the main source of this error comes from near misses and false positives.

Qualitative Evaluation. Yle’s professional editors have a posteriori evaluated the automatically generated segmentation. They generally observe that AI-based content segmentation is not yet able to produce sufficiently accurate results. When the editors looked at the timecode data generated by the system, the results included a large number of detected segment boundaries that did not actually represent the start or endpoint of a segment, but that are typical intra-scene breaks in a TV program, such as gaps in the music track, pauses in conversation, change of a spoken language, switch of camera angle, etc. Overall, the automatically provided timecode data for the two programs Strömsö and Urheiluruutu could not be directly utilised as a guide for publishing the program’s chapter markers which still has to be determined manually. Future evaluations need to be conducted in order to precisely judge the help that such technologies bring for semi-automatically adjusting the segment boundaries.

We observe better segmentation results on INA’s programs, in particular, when the program genre is news. Conversely, when the program’s genre is a sport magazine, the results are again disappointing revealing once more the difficulty of performing an adequate automatic content segmentation when there is a strong content coherency (e.g. the entire program is about sports).

2.4 Distant-Supervised Multimodal Content Segmentation

While Section 2.3 has described a fully unsupervised method for performing content segmentation, we describe in this section a method that leverages on an existing description of the content as a helper for segmenting the content. We therefore define the **Content Alignment** task as the goal of finding segment boundaries within the video content of a program while being provided with a human generated description of the program content which does not contain any timecode. In the case of MeMAD, these human generated content descriptions are typically a short description (*leads*) or a producer summary. The goal of this task is to aid the segmentation by providing a time reference to segments corresponding to content description.

2.4.1 Method

The pipeline used for content alignment is very similar to the segmentation pipeline (Figure 7), except that we do not leverage on any visual content. The similarity matrix is computed for

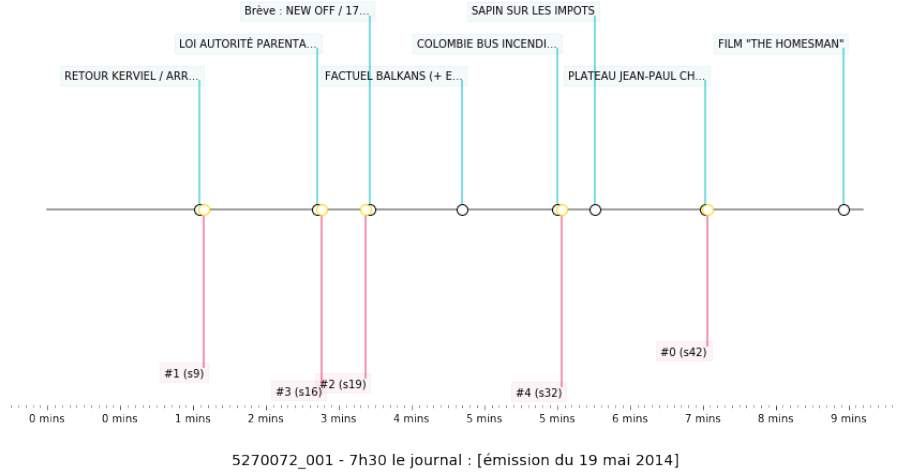


Figure 9: Segmentation results on a sample from INA dataset

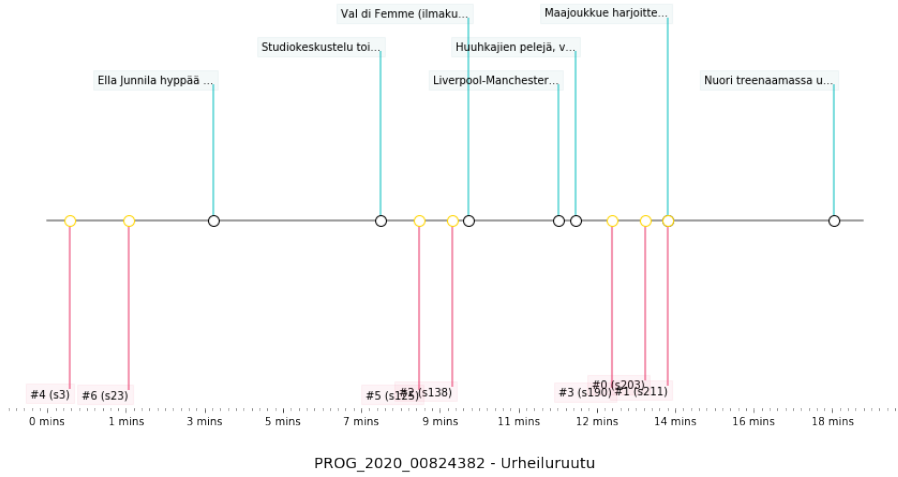


Figure 10: Segmentation results on a sample from Ulheiluruutu dataset

each pair of subtitle-description. Then, the segment boundaries are proposed to match the regions (defined by a sliding window) based on the similarity scores. The lowest succession of scores, i.e. where the description is very dissimilar to the content of multiple sentences, is suggesting a change in topic.

2.4.2 Evaluation

For evaluation, we use again the same INA-44 dataset which does not only contain the ground-truth segmentation of the content, but also content descriptions in the form of segment titles. We use the same metrics for evaluation, and we compare the results with the unsupervised content segmentation method described above (Section 2.3).

Method	P_k	$WindowDiff$
Content Segmentation	0.39	0.54
Content Alignment	0.35	0.51

Table 4: Content alignment results on subsets of the INA dataset

We see that the scores obtained are better (closer to 0) than those relying solely on the

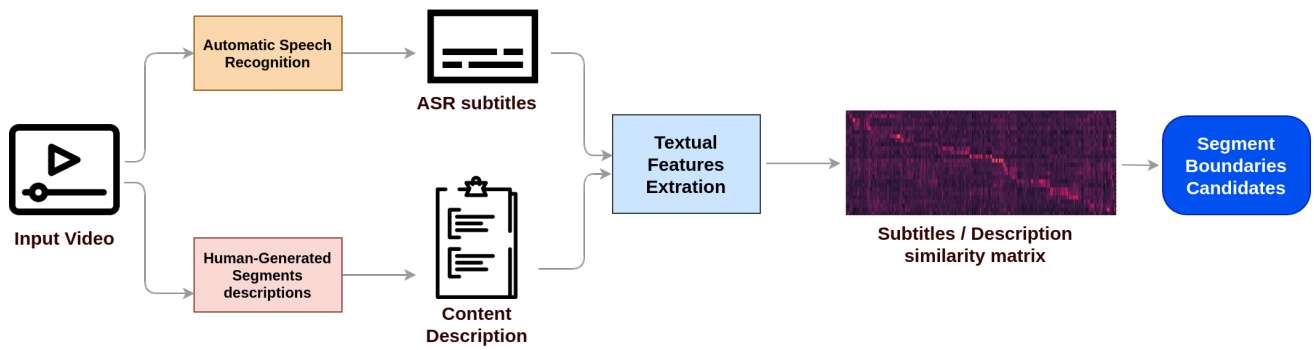


Figure 11: The content alignment pipeline (v1)

content of the subtitles themselves (Section 2.3). The improvement is however not very significant. There is still further development to be made in researching better ways to map content descriptions to program content and most notably, in providing representations that are better suited for this task instead of using a generic pre-trained multilingual language model such as Sentence-BERT. Other methods of pre-processing, aggregating, and scoring similarity can be used and combined to further improve the results. There is also a general question as whether this is always possible or not to establish a correspondence between a content description authored by a documentalist with what is actually said in the program.

3 Moments Enrichment

In this section, we explore various ways for enriching TV moments. First, we propose new methods for extracting and predicting topics that can be attached to moments (Section 3.1). In particular, we describe ToModAPI, a library and a RESTful API that implements numerous topic models and evaluation metrics. We compare the performance of numerous topic models and we demonstrate that no one is superior to the others in any condition. We also present ZeSTE, a novel method that leverages on the ConceptNet common sense knowledge graph in order to predict the topics of a document in a zero-shot fashion while providing an explanation for this prediction. We evaluate this method on the MeMAD corpora.

Second, we continue to investigate methods for extracting and disambiguating named entities mentioned in media content (Section 3.2). The disambiguation is performed with respect to Wikidata, thus providing encyclopedic enrichment to the programs. We propose a novel method named GraphNER that proposes to use Graph Convolutional Network to extract named entities as opposed to the traditional top performing methods using Bi-LSTM networks and CRF. We also investigate how to extract named entities directly from the speech without using the textual modality. Finally, we evaluate the performance of the named entity disambiguation tools on the MeMAD corpora and we contribute to the field by providing a new ground truth for the Finnish language.

Third, we propose a novel method for generating summary which are character centric (Section 3.3). In particular, we propose to leverage on fan-made content to drive the creation of the summary which is seen as the last type of enrichment we propose. We have evaluated our method on the BBC East Enders TV series as part of the TRECVID VSUM benchmarking campaign where we ranked 1st.

3.1 Extracting and Predicting Topics

3.1.1 TOMODAPI: A Topic Modeling API to Train, Use and Compare Topic Models

From LDA to neural models, different topic modeling approaches have been proposed in the literature. However, their suitability and performance is not easy to compare, particularly when the algorithms are being used in the wild on heterogeneous datasets. In [22], we introduced ToModAPI (*Topic MOdeling API*), a wrapper library to easily train, evaluate and infer using different topic modeling algorithms through a unified interface. The library is extensible and can be used in Python environments or through a Web API. ToModAPI allows to have a unified environment and protocol for comparing topic models, making use of a common pre-processing and the same implementation for evaluation. In particular, the library provide evaluation functions which rely on intrinsic metrics – different types of coherence, including a word-embedding-based one – as well as on ground-truth ones – homogeneity, completeness, v-measure [23].

The library is available as open source at <https://github.com/D2KLab/ToModAPI/>. A full paper describing the library has been published at the NLP-OSS 2020 workshop colocated with EMNLP 2020 (Annex B.3).

3.1.2 Watch Your Model: A Systematic Evaluation of Topic Models

Thanks to the unified framework ToModAPI, we empirically evaluate the performance of 9 topic models from the literature on different settings reflecting a variety of real-life situations in terms of dataset size, number of topics, and distribution of topics, using both metrics that rely only on the intrinsic characteristics of the results (coherence), as well as the agreement

between the resulting topic distribution and their ground truth. Our findings reveal some shortcomings regarding the common practices in topic models evaluation.

The results (Figure 12) reveal several differences between the trained models, which obtain more or less better performances in specific settings. Among these, LDA proves to be the most consistent resulting coherence, whereas the other algorithms excel in particular contexts and can be specifically suitable for a given dataset. Embedding-based models are particularly interesting because they prove to be less prone to generate meaningless topics. Increasing the number of topics is particularly helpful on bigger datasets, as it allows the topic models to find smaller yet more coherent subtopics within the collection, avoiding the drawback effect of being too specific.

We submitted a full analysis to ACL 2021 (Annex B.4).

3.1.3 Towards Zero-shot Explainable Topic Categorization Using a Common Sense Knowledge Graph

Assigning topical descriptors to media content is an essential task for understanding it on a high level. Yet this task requires collecting and annotating domain-specific and language-specific data to train classifiers to disambiguate between the different topics, and as a result, when adding or changing the list of supported topics, one must collect new data and train a new classifier.

We propose ZeSTE (for **Z**ero-**S**hot **T**opic **E**xtraction), a novel approach for topic categorization based on leveraging ConceptNet, a common-sense knowledge graph, to find terms related to labels of interest and perform zero-shot classification: i.e. without access to any labeled documents. Because the classification is based on the knowledge graph content, we can generate an explanation to the classifier decision based on the nodes of the graph that are relevant to the chosen label. Figure 13 shows the high-level classification pipeline for the method.

ZeSTE is publicly deployed at <http://zeste.tools.eurecom.fr/> while the code has been open sourced at <https://github.com/D2KLab/ZeSTE>. A paper describing and evaluating ZeSTE has been submitted to the LDK 2021 conference (Annex B.9).

3.1.4 Zero-shot Theme Extraction on MeMAD corpora

To demonstrate the use of the aforementioned method, we try to simulate the case where we have unlabeled data and we want to automatically tag it with its corresponding topic. To do so, we select a subset of INA programs that has already been annotated with *Themes*, which are close in principle to topics. These correspond to 80 programs tagged with 11 topics (labels in French): Cinéma, Humour, Information, Littérature, Musique, Politique, Religion, Société, Sport, Travail, and Variété.

As we can see in Table 5, for some labels such as Sport, Politics and Religion, the model is pretty accurate when relying solely on the subtitles (or ASR) of the programs. This is due to the fact that for these themes, the vocabulary used is strongly related to the label itself (e.g. all discussions about sports inevitably mention terms related to sport). Whereas for the other labels such as "Information", "Humor", "Society", "Work" and "Entertainment", the content of the programs is not necessarily related to the label itself. The surprisingly bad scores on Music, upon inspection, is mostly due to the errors coming from the automatic transcription of the programs.

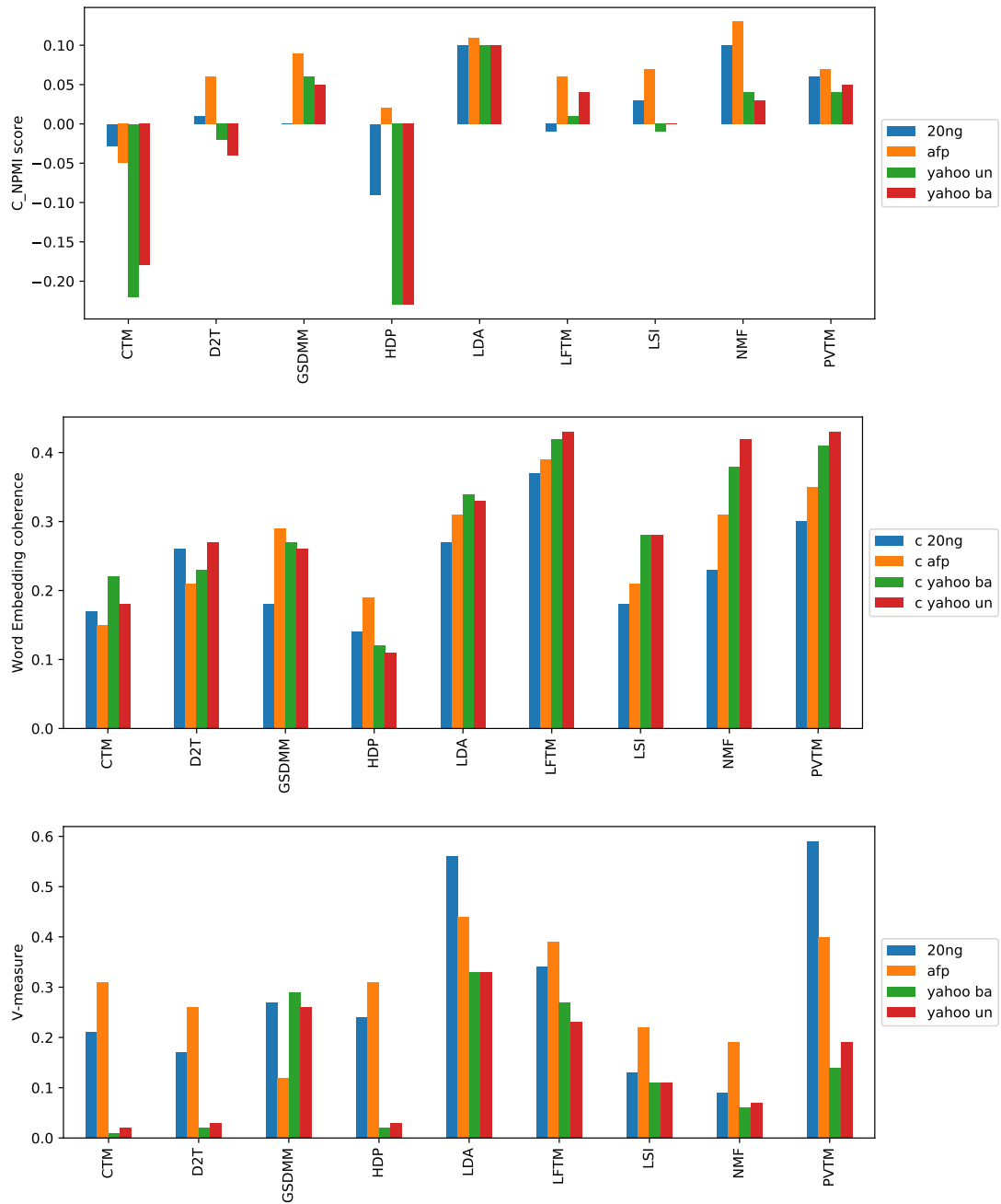


Figure 12: NPMI, Word embedding coherence and V-measure across the models trained on different datasets

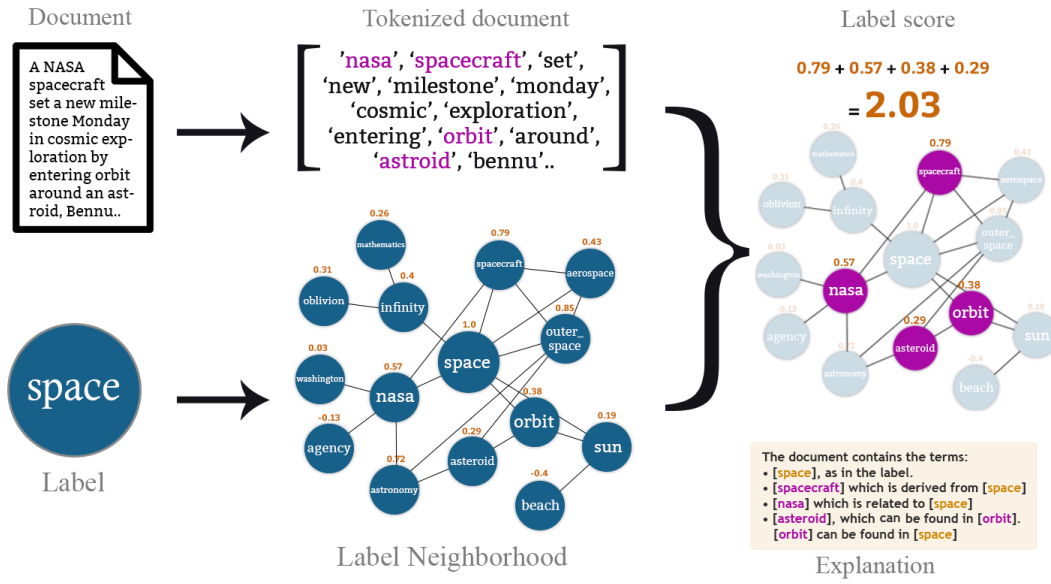


Figure 13: Illustration of the Zero-Shot topic categorization method (relation labels are omitted from the Label Neighborhood for clarity). Each node on the graph is associated with a score that corresponds to the cosine similarity between the graph embeddings of that node and the label node. We use the overlap between the document terms and the label neighborhood to generate a score for the label, as well as a natural language explanation for the prediction. We do so for all labels, and pick the one with the highest score.

3.2 Named Entity Recognition (NER) methodologies

3.2.1 GraphNER

Injecting real-world information (typically contained in Knowledge Graphs) and hand-crafted features into a pipeline for training end-to-end Natural Language Processing models is an open challenge. In this study, we propose to approach the task of Named Entity Recognition, which is traditionally viewed as a *Sequence Tagging* problem, as a *Graph Classification* problem. Instead of viewing each word in the sentence as a token in an ordered sequence, we model it as a node in a graph that links it to other words from its context (its neighborhood in the sentence), classes from external knowledge graphs (such as “Person first name”, “Company”, or “Capital”), as well as other properties that are known to be relevant for the task such as grammatical tags – all represented into nodes and fed as inputs to a classifier. We experiment with a variety of graph modeling techniques, and we evaluate our approach on the referenced CoNLL-2003 dataset⁶. Our results show that it is a promising direction towards integrating external knowledge and human expertise into the dominant end-to-end training paradigm.

Approach. By casting Named Entity Recognition as a graph classification task, we provide as an input to our model a graph representing the word in the training or the evaluation corpus that we want to tag (the *central node*), as well as its *context* – words appearing before and after it – and its *tags* (properties such as appearing in gazetteers, grammatical role, etc.), and we output the entity type, as seen in Figure 14. This formalization allows, in theory, to represent the entire context of the word (as graphs can be arbitrarily big), to explicitly model the left and the right context independently, and to add different descriptors (tags) to each word seamlessly (either as node features or other nodes in the graph) and thus help the model to leverage knowledge from outside the sentence and the closed training process. This graph is then embedded into a fixed-length vector and is fed to a classifier to predict the entity type.

⁶<https://www.clips.uantwerpen.be/conll2003/ner/>

label	Precision	Recall	F1-Score
sport	100.0	100.0	100.0
politique	77.3	70.8	73.9
religion	62.5	83.3	71.4
littérature	66.3	66.7	66.5
humour	60.0	42.9	50.0
information	38.9	70.0	50.0
musique	28.6	50.0	36.4
travail	100.0	100.0	18.2
société	0.0	0.0	0.0
variété	0.0	0.0	0.0

Table 5: Performance evaluation on INA data for a variety of themes

While we posit that this method is flexible and can integrate any external data in the form of new nodes or node features in the input graph, we focus on the following properties that are known to be related to the NER task:

- **Context:** which is made of the words around the word we want to classify.
- **Grammatical tags:** we use the Part of Speech tags (POS) e.g. ‘Noun’, ‘Verb’, ‘Adjective’, as well as the shallow parsing tags (chunking) e.g. ‘Verbal Phrase’, ‘Subordinated Clause’ etc.
- **Case:** the presence of uppercase letters usually signify that a word refers to an entity. We thus add the following tags: ‘Capitalized’ if the word starts with a capital letter, ‘All Caps’ if the word is made of only uppercase letters, and ‘Acronym’ if the word is a succession of uppercase letters and periods.
- **Gazetteers:** we generate lists of words that are related to potential entity types such as “Person First Name” and “Capital” (this is further explained in the next subsection).

Graph Representations. The literature on graph representations is diverse and provides a very large space for model exploration. For our experiments, we choose one instance of each of the several widely-used representations: a shallow neural auto-encoder, Node2Vec for node embeddings, TransE for entity embeddings, and a GCN (Graph Convolutional Network) based on [24]. We also train a two-layers neural network on a simple binary embedding of graph nodes as a baseline.

The challenge of representing graphs does not end there, as we can materialize the idea expressed so far in multiple ways:

- What constitutes the nodes of the graph and what can be modeled as a feature of the said nodes?
- How to connect these nodes? Should everything be connected to the central node or should the connection reflect the order in the sentence? Should these relations be semantic, i.e. of different types?
- Should we account for the entire context of the word or just limit it to a fixed-size window, and if so, what should be this window size?
- What is the direction of information propagation through the graph?

All of these design decisions (some are featured in Figure 15), on the surface, do not seem to have straightforward answers. We detail some of the choices in the experiments section.

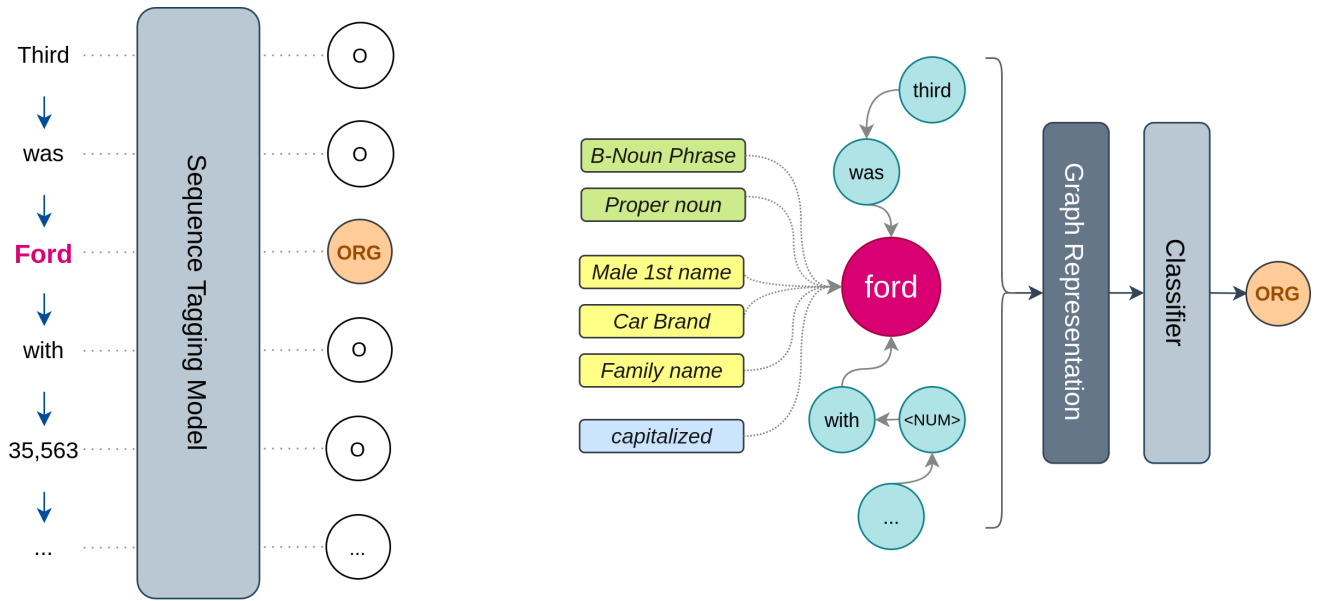


Figure 14: NER as graph classification: instead of the traditional sequence tagging model (left side), we propose to treat each word in a sentence as a graph where the word to classify is linked to the words from its context, as well as other task related features such as grammatical properties (in green), gazetteers mentions (in yellow) and task-specific hand-written features (in blue). The graph is turned into a fixed-length vector which is then passed to a classifier to predict the word label.

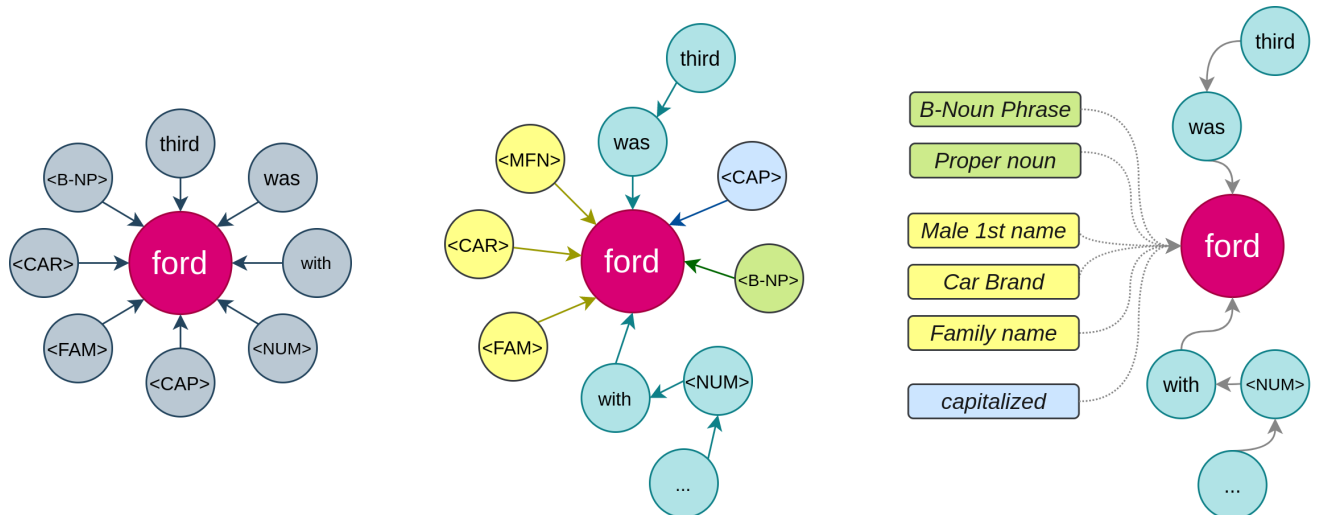


Figure 15: Several potential representations of word graphs: (a) every word in the vocabulary and every potential tag are nodes that are directly linked to the central node (b) the context nodes are connected in the same order as they appear in the sentence, and the relations to the node are explicitly differentiated (as seen by the color of the edges) (c) the same representation but with the tags added as node features to the central node, not as nodes themselves, i.e. only words are modeled as nodes in this representation

Experiments and Results. In this section, we detail the experiments we performed by training our model on the CoNLL-2003 training dataset and report the results obtained on its evaluation set. Unless specified otherwise, we consider the representation labeled as (a) in Figure 15, i.e. every word and every tag is a node in the graph, and they are all directly linked to the word to classify. To express the fact that different nodes relate to the central word with different relations, we concatenate their representations separately. Thus, the graph representation would be the concatenation of the individual representations of each type of relation (represented by different colors in the Figure 15). In case multiple nodes are attached to the central node with the same relation, we average their representations. For all training methods, we consider a context size of 3 (i.e. 3 words to the left and 3 words to the right of the central word), we use ReLU as the activation function between layers, and for all classifiers, we add weights to the loss function to accommodate for the unbalance in label distribution based on this formula:

$$w_{label_i} = \sqrt{\frac{\min(count(label_j) \text{ for } label_j \text{ in labels})}{count(label_i)}}$$

We classify each word in the corpus into one of the 5 entity classes and we report on the Accuracy, Micro-F1 and Macro-F1 scores for all trained models in Table 6. We note that the difference between Micro-F1 and Macro-F1 score is due to the over-representation of the "O" label in the dataset, as the Macro-F1 score averages the F1 score on each class regardless of its frequency, which brings the results down if the models do not perform equally well on all classes.

Binary Embedding baseline. For this model, we represent the graph as a binary embedding of the different nodes that are present in it. Concretely, we concatenate a one-hot embedding of the word, its left context and right context separately (multiple words can be present based on the size of the context we want to consider), and one-hot embeddings for all other extra tags in the vocabulary (e.g. gazetteers classes, POS tags, etc.). This binary representation is then fed into a 2 layers feed-forward neural network to predict the label of the word. In Table 6, Binary refers to the binary representation containing only the word and its neighborhood, Binary+ adds POS, CHUNK and Case tags, and Binary++ adds gazetteers tags as well. This later variant is the one which performs the best.

Binary Auto-Encoder. Using the same representation as Binary++, we first train a neural encoder-decoder (both 2 layers neural networks) to reconstruct the input binary representation of the graph. We then use the encoder part to generate a fixed-length vector (embedding) that is fed to a 2 layers feed-forward neural network to predict the label. We experiment with multiple dimensions for the embedding and report the results in Table 6. We can see that increasing the dimensionality of the embedding space (from 100 to 500 to 1000) improves the results accordingly, but the performance is severely lower than the model that is trained end-to-end with the binary representation.

Node Embeddings. We use Node2Vec⁷ to generate embeddings of different dimensions for all nodes in our graphs (including tag nodes). The results, as reported in Table 6, show that increasing the size of the embeddings does not significantly improve the results. We note again that this method does not account for the different node types as context nodes and tag nodes are all modeled similarly.

Graph Convolution Network. For this approach, we directly feed the graph data into a GCN (without pre-computing some embedding for the graph). We base our model on GraphSAGE-GCN [24], and we use the architecture based on this model from the PyTorch Geometric Library⁸ that we modify to account for additional node features and multi-class classification.

⁷<https://snap.stanford.edu/node2vec/>

⁸https://github.com/rusty1s/pytorch_geometric/blob/master/examples/proteins_topk_pool.py

The architecture is detailed in Figure 16.

We report on two variants: GCN in which nodes are only characterized by their value (the word itself or the tag), and GCN+, in which we append tags as one-hot features for the central node (similarly to representation (c) in Figure15). Unlike the previous methods where we linked all nodes to the central node, we link words to each other in the same order they appear in the sentence, so that order is accounted for when propagating information through the graph convolution and aggregation. In Table 6, we see that including the extra features into the node representation notably improves the results.

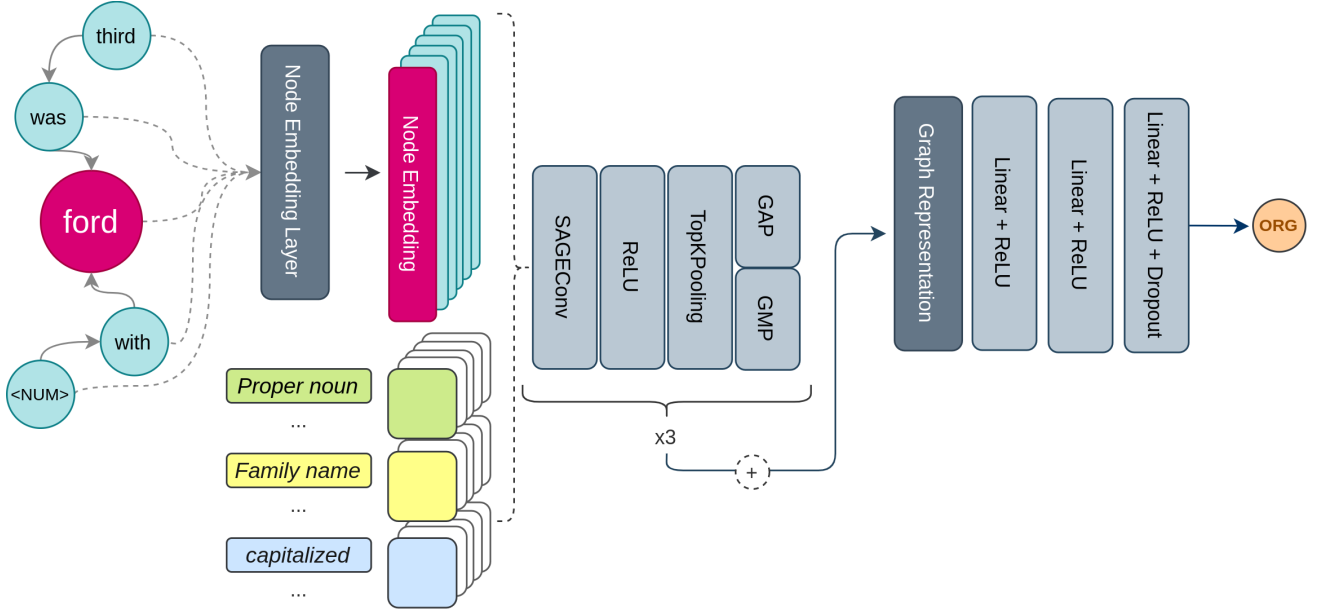


Figure 16: The Graph Convolutional Network architecture (GCN+)

Results. We also report the results of the best model from each family of graph representations on the test set together with the currently best performing approach (LUKE) in Table 7. Generally, we notice a sharp drop in performance for all models between the development and test datasets (especially Node2Vec), which is probably due to the fact that the test set contains a lot of words that do not appear in the training set (and thus get the $< UNK >$ generic representation).

Method	Accuracy	Micro-F1	Macro-F1
Binary	91.0	90.7	77.9
Binary+	94.4	94.2	81.9
Binary++	94.3	93.8	82.3
Auto-encoder-100	87.2	86.7	57.6
Auto-encoder-500	90.4	89.9	68.3
Auto-encoder-2000	91.8	91.5	71.7
Node2Vec-300	93.8	94.1	82.0
Node2Vec-500	93.8	94.1	82.5
Node2Vec-1000	93.8	94.1	82.1
GCN	96.1	96.1	86.3
GCN+	96.5	96.5	88.8

Table 6: Results of different graph representations on CoNLL-2003 evaluation set

Method	Accuracy	Micro-F1	Macro-F1
Binary++	92.1	91.4	76.8
Auto-encoder-2000	91.8	91.5	70.4
Node2Vec-500	90.2	91.1	72.6
GCN+	94.2	94.1	81.0
LUKE [25]			94.3

Table 7: Results of different graph representations on CoNLL-2003 test set

While the method proposed in this work shows some promising results, the performance on the test set is significantly lower (13.2 macro-F1 score drop) than the best state-of-the-art Transformer-based method as of today. This makes the approach, despite its theoretical potential, unusable in its current state.

As we expressed before, multiple design choices were made to limit the design space of models to experiment. Furthermore, it is known that hyper-parameters tuning can play a considerable role in performance and this is not yet exhaustively done for most methods, which leaves the possibility that different design choices and further tuning could lead to better performances overall.

We have open sourced the implementation of GraphNER at https://github.com/Siliam/graph_ner. A paper presenting this approach has been accepted at ESWC 2021 for the Poster Session (Annex B.5).

3.2.2 Spoken NER

The conventional way of doing NER from speech is through a pipeline approach where first an ASR system produces the transcripts and then a NER system annotates the transcripts with named entity tags. In such case, both systems are trained independently of each other, resulting in the ASR system not being optimized for the NER task and vice versa.

In this work, we present two approaches for doing named entity recognition from speech in an end-to-end manner, where one system generates the transcripts and annotates them with named entities. Both approaches are implemented using an attention-based encoder-decoder architecture (AED). The first approach is called augmented labels (AL) and in this approach during training, the original transcripts are augmented with named entity tags, such that each word is followed by its corresponding tag. That way the system will learn to produce transcripts that are annotated with named entities. The second approach is called multi-task (MT) and it is an attention-based encoder-decoder architecture, similar to the augmented labels approach. The difference between them is that in the multi-task approach, there are two decoder branches, one for doing ASR and one for NER. Additionally, in this approach, we are using the original transcripts and not the augmented ones.

Data. In our experiments, we used four different data sets in three different languages: English, Finnish, and Swedish.

English data. Even though the goal of the work was to do NER on low-resource languages, we wanted to additionally test our models on a well-known language, like English.

For the English experiments, we used the whole LibriSpeech data set [26], consisting of about 1000 hours of recordings. Since this data set is not annotated with gold-standard named entity tags, we used a separate NER system to annotate the transcripts. The NER system that was used for annotation is described in more detail later in the work. For testing the system on a gold-standard named entity recognition data, we used a data set which is a subset of a

combination of multiple ASR data sets, such as CommonVoice, LibriSpeech, and VoxForge. We will call this data set English-Gold. The English-Gold data set is annotated and provided by Hemant et al. [27]. The number of tokens and named entity tags in the English data sets are presented in Table 8.

Parameters	LibriSpeech	English-Gold
Audio length	1000 h	148 h
Total tokens	9.6 M	1.3 M
Unique tokens	87600	41379
PER tags	194172	50552
LOC tags	66618	23976
ORG tags	11415	5025

Table 8: Data distribution for the LibriSpeech and English-Gold data sets.

Swedish data. For the Swedish experiments, we used the Sprakbanken corpus, which is a public domain corpus, hosted by the National Library of Norway. The data set consists of 259 hours of recordings. Since the data set does not have gold-standard named entity tags, we used a separate NER system to annotate the transcripts. The NER system that was used to obtain the tags is described in more detail later in the work. The number of tokens and named entity tags are presented in Table 9.

Parameters	Count
Audio length	259 h
Total tokens	1.4 M
Unique tokens	69310
PER tags	23258
LOC tags	7585
ORG tags	2231

Table 9: Data distribution for the Swedish data set.

Finnish data. For the Finnish experiments, we used the Finnish parliament data set [28], consisting of about 1500 hours of recordings from the Finnish parliament. Since we do not have gold-standard named entity annotations for this data set, we used a separate NER system to obtain them. The NER system that we used to annotate the transcripts is explained later in the work. The number of tokens and named entity tags in the data set are presented in Table 10.

Parameters	Count
Audio length	1500 h
Total tokens	7.3 M
Unique tokens	337423
PER tags	44984
LOC tags	73860
ORG tags	65463

Table 10: Data distribution for the Finnish parliament data set.

NER systems for annotating the ASR data sets. For annotating the English LibriSpeech data set, we used the large uncased BERT model [29], which we fine-tuned on the CoNLL 2013 data set [30]. To obtain the named entities for the Swedish ASR data, we used the Swedish BERT model [31], which was already optimized for the NER task.

For annotating the Finnish parliament data set with named entity tags, we used a NER system that was developed at Aalto University. The system uses a bidirectional LSTM (BLSTM) neural network [32] with a Conditional random field (CRF) [33] on top. The architecture utilizes word, character, and morph embeddings. The architecture is explained in more detail in [34].

Pipeline NER systems. To see how the pipeline approach of first generating the transcripts using an ASR system, and then annotating them with a NER system performs in comparison to our proposed models, we trained BLSTM-CRF models for each of the data sets. The architecture and the parameters are identical to the NER branch in the multi-task approach, which is explained later in the work.

Augmented labels approach. The augmented labels approach uses an attention-based encoder-decoder architecture, which takes as input audio features and outputs the corresponding transcripts, annotated with named entity tags. The encoder is a BLSTM neural network, that takes the audio features, in our case log filter banks, and compresses them in a hidden vector representation that is passed to the decoder. The decoder is an LSTM neural network that is initialized using the hidden vector representation. The job of the decoder is to produce the transcripts, annotated with named entity tags. It does that using the Luong attention mechanism [35], where as scoring function, we used hybrid + location-aware, as described in [36]. In the experiments where we additionally used the CTC loss function [37], the final loss was calculated as:

$$L_{asr} = \lambda L_{ctc} + (1 - \lambda) L_{aed} \quad (2)$$

where, L_{ctc} is the CTC loss, L_{aed} is the decoder loss and λ is the weighting factor that determines the contribution of the separate loss functions to the final loss.

As true labels, in this approach we used the original transcripts, augmented with named entity tags, in a way that each word is followed by its corresponding tag. That way, the system can learn to generate transcripts annotated with named entity tags. A sample output from this approach is shown in Figure 17.

Lucas	PER	lives	O	in	O	Canada	LOC
-------	-----	-------	---	----	---	--------	-----

Figure 17: Augmented labels output.

Multi-task approach. The multi-task approach is an attention-based encoder-decoder architecture, similar to the augmented labels approach. The difference between them is that in this approach we have two separate decoder branches. The first branch does the ASR and is identical to the one in the augmented labels approach, whereas the second one does the named entity tagging, and it consists of a BLSTM with a CRF layer on top.

Since we have two decoder branches, two loss functions need to be jointly optimized. The final loss function is calculated as:

$$L = \beta L_{asr} + (1 - \beta) L_{ner} \quad (3)$$

where L_{asr} is the loss from the ASR decoder, L_{ner} is the loss from the NER decoder, and β is a weighting factor that determines the contribution of both loss functions.

Similar to the augmented labels approach, in the experiments where we utilized the CTC loss, the ASR loss L_{asr} is calculated as in Equation 2. Unlike the augmented labels approach, which outputs combined transcripts and NER tags, in this approach, we have two separate outputs, as shown in Figure 18.

Lucas	lives	in	Canada	PER	O	O	LOC
ASR transcript				NER tags			

Figure 18: Multi-task output.

Experiments. In all of the experiments, we used logarithmic filter banks with 40 filters as features. As an optimizer, we used Adam [38] and negative log-likelihood as a loss function. In the multi-task approach, when the model converged, we froze the encoder and the ASR decoder branch and additionally trained the NER branch, which improved the results in most of the data sets. We will refer to this model as MT*.

Finnish experiments. The audio features tend to have big lengths, so processing them with a standard BLSTM can be computationally challenging. For that purpose, we used a pyramidal BLSTM, which reduces the time resolution by half in every consecutive layer. In our case, the encoder consists of 5 pyramidal BLSTM layers with a hidden size of 300. A dropout of 0.1 is applied after the pyramidal BLSTM network.

In the augmented labels approach, the decoder has a character embedding layer of size 150, followed by a single-layer LSTM network with a hidden size of 300 and a hybrid + location-aware attention size of 300. The number of filters in the location-aware convolution part is 100. A dropout of 0.1 is applied after the attention mechanism.

In the multi-task approach, the ASR decoder is identical to the one in the augmented labels approach. The NER decoder uses 300 dimensional pre-trained fastText word embeddings [39], which are used as an input to the one-layer BLSTM network with a hidden size of 300. The BLSTM network is followed by a fully-connected layer with a hidden size of 300, and a dropout with a probability of 0.1. The output of the fully connected layer is passed through a CRF layer which produces tag probabilities. To combine the separate loss functions, as in Equation 3, we used a β weighting factor of 0.8

Swedish experiments. The Swedish data set consists of short utterances, so we decided to use 3 normal and 2 pyramidal BLSTM layers in the encoder, with a hidden size of 450, followed by a dropout layer with a probability of 0.1.

In the augmented labels approach, the decoder has a character embedding layer with a size of 150, same as in the Finnish experiments, followed by a BLSTM network with a hidden size of 450. The number of filters in the location-aware convolution element, in this case, is 150. A dropout of 0.1 is applied after the attention mechanism.

In the multi-task approach, the ASR decoder is identical to the one in the augmented labels approach. The NER decoder uses 300 dimensional fastText word embedding, just like in the Finnish experiments. The embeddings are passed through a one-layer BLSTM network with a hidden size of 450, followed by a fully-connected layer with a size of 450 and a dropout of 0.1. In the end, the output is passed through a CRF layer which produces the tag probabilities. The β weighting factor is the same as in the Finnish experiments. In both the augmented labels and the multi-task approach, we additionally added the CTC loss function and combined it as in Equation 2, with a λ weighting factor of 0.2.

English experiments. For the LibriSpeech data set, the parameters are almost identical to the ones in the Swedish experiments, with the only exception being that all the layers use a pyramidal BLSTM structure in the encoder. Since the English-Gold data set is relatively small and not sufficient to train a separate system on it, we used the pre-trained model on the LibriSpeech data set and fine-tuned it on this data set.

Results. The pipeline models that we constructed are evaluated on the transcripts generated by the multi-task models, trained on each of the data sets. In Table 11, we can see how each of the models performs in terms of precision, recall, and F1 score.

In Table 12, we can see how the multi-task and the augmented labels approach perform on the Finnish test set when evaluated on the transcripts that they produced. From the results, we can see that the multi-task approach where we additionally fine-tuned the NER branch outperforms the other two models. Additionally, we can see that both multi-task approaches

Model	Precision	Recall	F1
Parliament (Finnish)	93.63	85.64	89.46
Swedish	69.35	79.37	74.02
Libri clean	76.43	79.09	77.74
Libri other	64.07	74.40	68.85
English-Gold	79.24	71.28	75.05

Table 11: Precision, recall and F1 score for the pipeline models.

perform better than the pipeline approach.

Score	AL	MT	MT*
Precision	92.65	93.35	93.17
Recall	81.61	87.80	88.80
F1	86.78	90.49	90.93

Table 12: Precision, recall and F1 score for the Finnish test set , where NER is done on the transcripts generated by the models.

In Table 13, we can see how both approaches perform on the Swedish test set when evaluated on the transcripts that they produced. From the results, we can see that in this case, the augmented labels approach outperforms the multi-task approaches. Also, we can see that both the augmented labels and the fine-tuned multi-task approach perform better than the pipeline approach, whereas the standard multi-task approach falls behind.

Score	AL	MT	MT*
Precision	74.96	70.14	74.19
Recall	78.13	77.94	76.67
F1	76.51	73.83	75.41

Table 13: Precision, recall and F1 score for the Swedish test set , where NER is done on the transcripts generated by the models.

In Tables 14 and 15, we can see how both approaches perform on the LibriSpeech and English-Gold test sets when evaluated on the transcripts that they produced. On the LibriSpeech test set, we can see that the fine-tuned multi-task approach performs better than the other two approaches. However, it is still slightly behind the pipeline approach. On the English-Gold test set, on the other hand, the standard multi-task approach performs better than the other two approaches. Additionally, it outperforms the pipeline approach by a large margin. Additionally, we can see that the augmented labels approach also outperforms the pipeline approach on this data set.

Model	Libri clean			Libri other		
	Pre	Rec	F1	Pre	Rec	F1
AL	79.77	63.47	70.69	70.21	52.15	59.85
MT	74.63	76.77	75.68	60.90	73.44	66.59
MT*	76.33	77.10	76.72	63.33	71.75	67.29

Table 14: Precision, recall and F1 score for the LibriSpeech test set, where NER is done on the transcripts generated by the models.

Score	AL	MT	MT*
Precision	82.60	77.04	81.86
Recall	69.30	84.89	68.02
F1	75.21	80.78	74.30

Table 15: Precision, recall and F1 score for the English-Gold test set, where NER is done on the transcripts generated by the models.

In the future, we plan to replace the BiLSTM encoder-decoder network with a Transformer model. Additionally, we plan to utilize the part-of-speech tags and see if they help in the NER task. Another thing worth exploring is the scheduled sampling, which might improve the results.

3.2.3 Evaluating Named Entity Linking (NEL) Approaches on MeMAD Data

Generating Gold Standard NEL Annotations. The Urheiluruutu data set contains in total 15 programs, split into a train set (134.6 minutes in 12 program episodes) and test set (19.3 minutes and 3 program episodes).

Table 16 shows the details of the data set including the number of the NEL labels. The data set was selected as it was easily available within the project, as it is also used in the content segmentation PoC described in D7.4. Since the sports program contains a lot of material from different rights holders, we cannot share the media outside the MeMAD project, but similar type of content from the MeMAD project will be opened (see details in D1.7).

	Audio length	Files	Total words	B-tags	ASR errors
Train set	134.6 min	12	14698	1127	354
Test set	19.3 min	3	2460	163	46

Table 16: The description of the the Urheiluruutu data set. The number of words is the total number of words in the ASR transcript of the programs

The annotation pipeline to create this set was the following: We first ran the Urheiluruutu audios through Lingsoft’s Finnish ASR. Then the ASR result was fed into Lingsoft’s LMC NEL pipeline which links the recognized Named Entities into Wikidata and is described in detail in the following. The output of this analysis was then uploaded into a BRAT rapid annotation tool [40]⁹. A Lingsoft expert corrected the NER results and the NEL links and tagged possible ASR errors in the named entities. No corrected transcript for the speech in the audio of each program was available, thus the expert resorted to tagging only the speech recognition errors in the named entities. The annotated data was output from the BRAT and converted into the IOB2 format with an open source tool¹⁰. Compared to NER IOB2 format, in the current case, the label for each entity is the correct Wikidata URI. The ASR error tags were added as an additional comments column.

Since there was no corrected transcription, care was taken to annotate the ASR errors that contributed to further NER/NEL errors. More specifically:

- When an ASR error clearly contributed to a NE being incorrectly linked, it was marked with <ASR> in the Notes field in Brat and added to the extra column in the IOB2 format
- If an ASR error in one NE directly contributed to other NEs being incorrectly linked, they were also marked with <ASR> even if the ASR result there was correct (e.g. there is an ASR error when introducing a person with their full name and subsequent mentions with just the last name are correct, but cannot be linked because the full name is erroneous/missing)
- In the case of recurring sports events, for example ice hockey world championships, the main wikidata entity is linked, not the entity of the individual event (e.g. world championships of year X).

⁹<http://brat.nlplab.org>

¹⁰<https://github.com/spyysalo/standoff2conll>

Lingsoft NEL pipeline. Lingsoft’s Named Entity Linking pipeline (LMC NEL) is described in the following. The components of the pipeline are shown in figure 19.

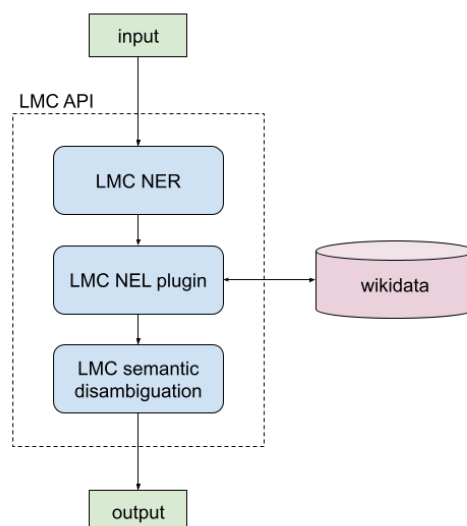


Figure 19: The components of the Lingsoft LMC NEL pipeline

The input text is first processed with the standard LMC NER, which is a rule-based tagger. In addition to classifying the spans into NE categories such as persons, locations etc., it also attempts to normalize different realizations of an entity (such as "Martti Ahtisaari", "president Ahtisaari" and "Ahtisaari" into a unique label. Examples of such normalized queries are given in Table 17. LMC NER has previously been described in deliverables D3.1 and D3.2. The NER analysis is then passed on to an LMC plugin that performs the Wikidata linking. The plugin generates a SPARQL query from the normalized labels from NER. Notice that there might be separate types of entities already recognized in the NER phase, such as the Liverpool FC football team and the city of Liverpool as in the third example.

Recognized entity		Normalized form
SDP:n	⇒	SDP (ORG)
Ahtisaaren	⇒	Martti Ahtisaari (PER)
Liverpoolin	⇒	Liverpool FC (ORG)
	⇒	Liverpool (LOC)

Table 17: Normalization of the recognized entities (type in parenthesis) with the LMC NER. Notice that partial matches can be matched to full name, if they are introduced fully as such elsewhere in the text.

In addition to the entity ID, a selection of properties (e.g. “instance of”, “occupation”, “coun-

try”, etc.) are also requested in the query. To reduce load a query cache is used. For each query, the query cache is checked, and only labels that have not previously been queried, are added to the query. The query is executed and the query cache is updated. If there are a lot of previously unseen labels, the query is split into smaller pieces in order to stay within the limits of accepted query size. Some adjustments are done to the query results to account for inconsistencies (e.g. entity exists, but is missing the “instance of” property, but has a “subclass of” property).

Due to the lack of Finnish training data for the LMC NEL, a heuristic for relevance is used. A baseline score is attached to the entities based on their type (the “instance of” property) so that certain kinds of entities can be preferred over others when there is no supporting evidence from the context. This is used to prefer e.g. cities and countries over names of music albums, which in practice have shown to be a very common source of false taggings. The amount of sitelinks is also added to this score to favor more “popular” entities. This has the effect of preferring Michael Jackson, the well-known singer over the many other Michael Jacksons in wikidata. The different properties returned for the entities are added as features to the candidate senses in LMC. These candidate senses are passed to the existing semantic disambiguation mechanism of LMC.

Semantic disambiguation is then done by preferring entities that have support elsewhere in the text. Here, support means that the entity has features that also exist in entities elsewhere in the text. This has the effect of interpreting for example “Florida”, (part of Florida Panthers) as an ice hockey team instead of a state in the United States, when there are mentions of ice hockey players or other ice hockey teams (entities having the property “sport” = “ice hockey”). Nearby support (in the same sentence) is weighed more than other support. The entity with the highest total score (baseline score + feature support score) is selected.

The Lingsoft NEL plugin might benefit from several changes to improve the recall and the disambiguation. These changes might also mean additional errors in both recall and precision, which is why they have yet to be implemented. For example, once the query term is normalized, it is queried only from the Finnish Wikidata. Querying in Wikidata in multiple languages, especially English would probably improve the recall, although it might introduce additional errors. Similarly, relaxing the requirement for the exact label match and allowing variations within a short edit distance might improve recall results. This relaxation would probably be most beneficial with foreign names especially when there is variation in the pronunciation and transliteration of person names. For example, the president of Belarus’s name is transcribed in English as Alexander Lukashenko, whereas the Finnish transcription is Aljaksandr Lukašenka.

Currently the information about the type of the Named Entity from the NER phase is not used in the query phase from Wikidata, but this could be included. This feature has been intentionally left out as the NER types encoded in the NER lexicon are a subset of all possible and new names, but it could be used as an additional feature for the decision making process. Similarly, there are many cases in which the only item found in Wikidata is not the correct one - for example there is an actor with a fairly common name but the text describes an athlete. At the moment, the incorrect one is returned, but it would be better to return a ‘NIL’ instead.

The relevance of the correctly returned entities is currently an unresolved question within the Lingsoft pipeline. Currently the NEL linking setup nor the evaluation data does account the relevance except based on the number of mentions, but types of entities are not weighted differently. An easy heuristic would be to adopt the finding from the EU project Linked TV [41], which found that ORG and PER entities are usually found more relevant than LOC and study this phenomenon further. In addition, training data that takes into account also the perceived relevance of each returned entity would be very beneficial.

Aalto NEL model. In the following section, we describe the Aalto NEL model. The building blocks of the model are presented in Figure 20.

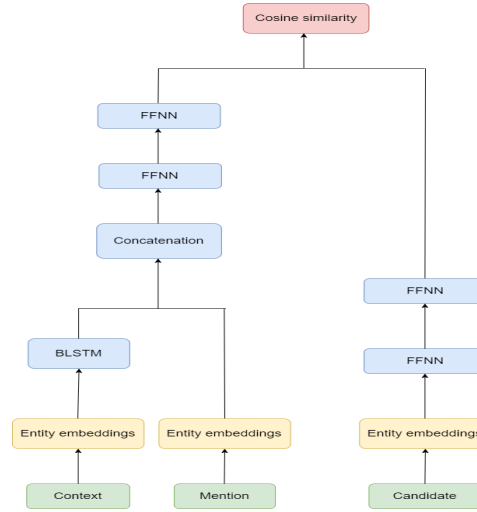


Figure 20: Aalto NEL architecture.

From the figure, we can see that the system uses three different inputs: context, mention and candidate. The context in our case is the text that surrounds the named entity for which we need to retrieve a Wikidata link. It typically contains the whole sentence, excluding the entity mention. The second input is the mention, which in our case is the named entity for which we need to retrieve the link. The third input is the candidate entity, which the system compares with the mention and determines if it is the right one. The candidate, besides the name, contains description and alias information, which are retrieved from Wikidata. In practice, we generate up to three candidates, from which the system chooses the most likely candidate one. To represent the inputs as a vector representation, we used 300 dimensional wikipedia2vec entity embeddings [42], trained on the Finnish Wikipedia dump. If the mention or the candidate contain more than one word, their embeddings are averaged.

The input context is processed through a one-layer BLSTM, after which, the output is concatenated with the embeddings of the mention, generating a context-mention pair. The context-mention pair is then passed through a fully-connected layer with a hidden size of 600 and a dropout layer with a probability of 0.1. The output of the fully-connected layer is passed through a ReLU non-linearity and finally passed through another fully-connected layer with a size of 600 and a dropout of 0.1.

The candidate entity is processed in a similar way as the context and mention, with an exception being that we do not use a BiLSTM network. The entity embeddings obtained for the candidate entity are passed through a fully-connected layer with a size of 600, after which a dropout layer with a probability of 0.1 is applied. The output is followed by a ReLU non-linearity and another 600 dimensional fully-connected layer and a dropout of 0.1.

At the end, the context-mention pair is compared with each of the candidates, using the cosine similarity score. The candidate with the highest score is chosen as the correct one. If all the similarity scores are below a threshold value of 0.1, then we assume that none of the candidates are correct.

During training, the model is optimized to give a higher score to the correct candidate, than the two other negative candidates. The correct candidate should have a score higher by a margin of 1, in comparison to the other two negative candidates. The loss function is defined as:

$$loss = \sum_{e'} \max(0, -1 * (sim(e, cm) - sim(e', cm)) + 1) \quad (4)$$

where, e is the true entity, cm is the context-mention pair, e' is the candidate entity, and sim is the cosine similarity function.

During the evaluation, the test data set is first passed through the Aalto NER system [34], then the NEL system links the detected entities to the corresponding Wikidata resources. Since the mentions are often conjugated, we used an open source lemmatizer [43], to get the mentions in their basic form, before retrieving the candidate entities from Wikidata.

In the future, we plan to use a different lemmatizer. Additionally, we plan to normalize the entities even further, which can help with generating better candidate entities.

Evaluation Results. The Urheiluruutu training set was used for training the Aalto model. For Lingsoft LMC NER, all of those rule-based changes that could be made were made into the system based on the training data. Additional ASR data of three programs was used for evaluation purposes. Table 18 gives the results for both the Lingsoft LMC NEL and the Aalto system either with ASR errors ignored or including them.

	precision	recall	F1
LMC NEL excluding ASR errors	0.8395	0.7047	0.7662
LMC NEL (all)	0.7684	0.5690	0.6538
Aalto NEL excluding ASR errors	0.5100	0.3333	0.4031
Aalto NEL (all)	0.4636	0.2698	0.3411

Table 18: Lingsoft LMC NEL and Aalto NEL micro average results. It must be noted that the Lingsoft LMC NEL is benefiting greatly from the rule based NER and normalization, whereas the Aalto system has been trained only with the available Urheiluruutu data and an open source lemmatization tool [43].

The results obtained by the Aalto NEL system are significantly worse than the LMC NEL one. One reason for that is because the Aalto NEL system is neural network-based and requires a lot more training data than what we currently have. Additionally, the Aalto NEL system uses an open source lemmatization tool, which is different from the one developed at Lingsoft. The Aalto system does not normalize the acronyms, which results in many mentions not having any candidate entities. The number of those mentions is 87 and they are treated as errors during the evaluation.

3.3 Video Summarization

Considering video summarization as an important task for digital content retrieval and reuse, the TRECVID [44] Video Summarization Task (VSUM) 2020 aims at fostering the research in the field by asking its participants to automatically summarize “the major life events of specific characters over a number of weeks of programming on the BBC EastEnders TV series”¹¹. More precisely, for three different characters of the series, the participants have to submit 4 summaries with respectively 5, 10, 15 and 20 automatically selected shots. These generated summaries are evaluated by the assessors according to their tempo, contextuality and redundancy as well as with regards to how well they contain answers to a set of questions unknown to the participants before submission. In addition to the videos, the episode transcripts are provided by the organizers.

We propose a character-centered content summary approach based on fan-written synopses. Our approach relies on fan-made content and, more precisely, on the BBC EastEnders episode

¹¹<https://www-nlpir.nist.gov/projects/tv2020/vsum.html>

synopses from its Fandom Wiki¹². This additional data source is used together with the provided videos, scripts and master shot boundaries. We also use BBC EastEnders characters' images crawled from the Google search engine in order to train a face recognition system. All our runs use the same method, but with varying constraints regarding the number of shots and the maximum duration of the summary. The shots included in the summaries are the ones whose transcripts and visual content have the highest similarity with sentences from the synopsis. The runs submitted are as follows:

- MeMAD1: 5 shots with highest similarity scores and the total duration of the summary is ≤ 150 sec;
- MeMAD2: 10 shots with highest similarity scores and the total duration of the summary is ≤ 300 sec;
- MeMAD3: 15 shots with highest similarity scores and the total duration of the summary is ≤ 450 sec;
- MeMAD4: 20 shots with highest similarity scores and the total duration of the summary is ≤ 600 sec.

3.3.1 Approach

Our fan-driven and character centered approach is presented in Figure 21.

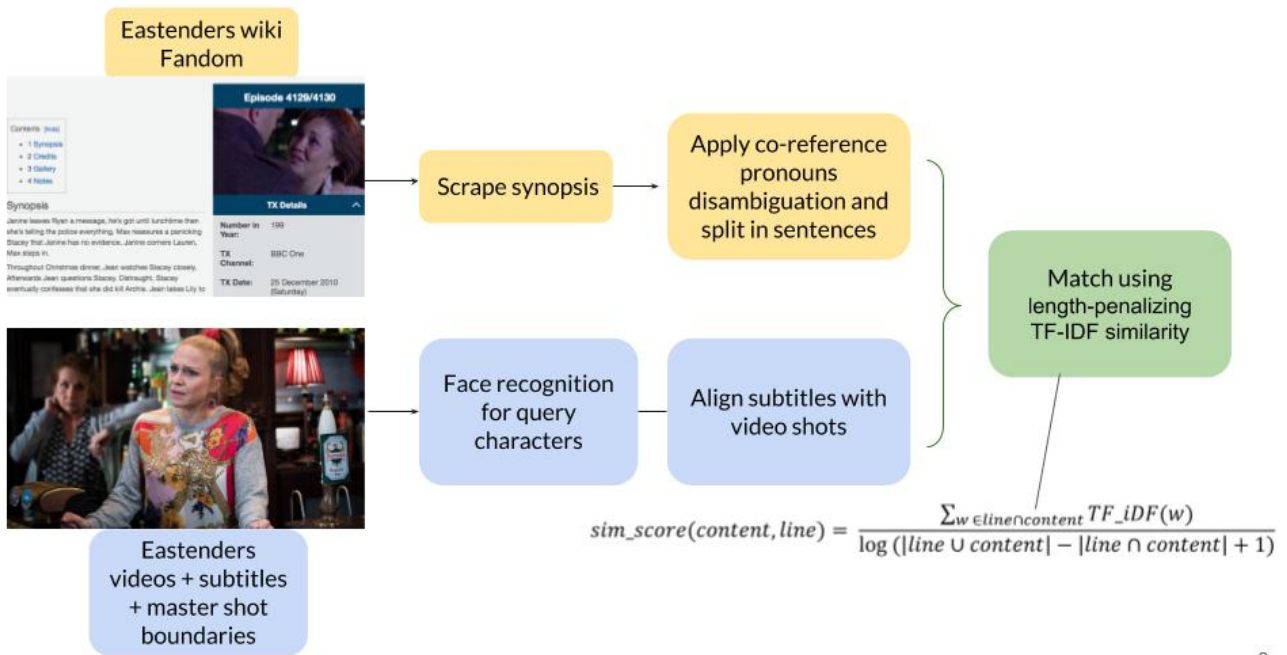


Figure 21: Fan-driven and character centered approach

Scraping Synopses From the Fandom Wiki and Selecting Shots. The first step of our approach consists in scraping synopses available on the Fandom EastEnders Wiki¹³. Our main hypothesis is that every sentence (ending with a period) represents an important event to be added

¹²https://eastenders.fandom.com/wiki/EastEnders_Wiki

¹³<https://eastenders.fandom.com/wiki/EastEndersWiki>

to the final video summary. We scrape the Synopsis and the Cast sections for each episode broadcasted between the dates of the provided episodes. The mapping between the episodes and their dates is in `eastenders.collection.xml` provided by the challenge organizers.

In parallel, we extract the shots in which the three characters of interest appear from the video. We run the Face Celebrity Recognition library¹⁴, a system that relies on pictures crawled from search engines using the actor's name as search keyword. In our experiments, we have added "EastEnders" to the character names in order to avoid retrieving pictures of different people with the same name. For each picture, faces are detected using the MTCNN algorithm and the FaceNet model is applied to obtain face embeddings. Following the assumption that the majority of faces are actually representing the searched actor, other faces – e.g. person portrayed together with the actor – are automatically filtered out by removing outliers until the cosine similarity of face embeddings has a standard deviation below a threshold of 0.24 which has been empirically defined.

The remaining faces are used to train a multi-class SVM classifier, which is used to label the faces detected in the frames. For more consistent results between frames, the Simple Online and Realtime Tracking algorithm (SORT) has been included, returning groups of detection of the same person in consecutive frames.

We select the shots displaying any of the the three characters of interests, keeping only those detections having a confidence score greater than 0.5. We also tried to use speaker diarisation to corroborate the visual information about the characters. However, given the limitations of the current technologies in terms of number of characters and the difficulty of identifying the character corresponding to each voice, we could not pursue the idea further.

Synopses and Transcript Pre-Processing. A synopsis for each episode was created using the provided files `eastenders.collection.xml` and `eastenders.episodeDescriptions.xml`. Since these were "EastEnders Omnibus" episodes, they correspond to multiple actual weekday episodes. We use the dates and the continuation to generate one synopsis for each "long" episode (typically made of 4 episodes). We then split the synopses into sentences and we perform coreference resolution on the synopses to explicit character mentions using <https://github.com/huggingface/neuralcoref>. In parallel, the provided XML transcripts were also converted into timestamped text and aligned with the given shot segmentation. Finally, both the synopsis sentences and shot transcripts were lower cased, stop words removed and lemmatized.

We also produced automatically-generated visual captions following the method presented by the PicSOM Group of Aalto University's submissions for the TRECVID2018 VTT task [45]. The hypothesis is that by describing the visual information of a shot, visual captions could complement well the dialog transcript and therefore allow for a better matching between the shots and synopses sentences.

Matching and Runs Generation. We perform a synopsis sentence / shot transcript pairwise comparison by generating a similarity score. We define similarity between two sentences as the sum of TF-IDF weights (computed on the transcript) for each word appearing in both of them, divided by the log length of the concatenation of both sentences, thus penalizing long sentences that match with many transcript lines.

Next, we order the shots by similarity score, picking only the best match for each shot (but not the other way around). This gives us scenes we are sure to appear in the summary, but not necessarily any guarantee about how important these scenes are. We also performed the pairwise comparison adding the automatically generated captions. A qualitative assessment

¹⁴<https://github.com/D2KLab/FaceRec>

revealed, however, that the captions were too noisy to complement the transcript well. We also make sure that if a line of dialog runs through the next shot, we include the next shot as well to improve the smoothness of the viewing. However, this heuristics was only relevant for the longest run (20 shots). Each run is made by selecting the N most matching shots out of the top, in chronological order.

3.3.2 Results and Analysis

The final results for the two teams which have participated in TRECVID VSUM are presented in Table 19 while the detailed scores of our approach are presented in Table 20. Our method obtains the best overall score for each of the 4 required runs. The mean scores (range 1 - 7. High is best) for tempo, contextuality and redundancy are all above average (respectively 4.75, 4.75, 4.1) despite the fact that our method does not specifically attempt to optimise these metrics. However, in terms of question answering, the results show that the shots selected did not allow to answer more than two (at best) of the five questions. More specifically, Table 21 shows (in bold) the questions that were answered in at least one of ours runs. We notice that most of the questions started either with 'What' or 'Who' and that our approach performed equally for both types of questions.

Table 19: Average score for each run and team

TeamRun	Percentage
MeMAD1	31%
MeMAD2	31%
MeMAD3	35%
MeMAD4	32%
NIIUIT1	9%
NIIUIT2	8%
NIIUIT3	8%
NIIUIT4	6%

Table 20: Detailed score for MeMAD's approach. The labels are made of the character name (e.g. Janine) followed by the run number (MeMAD 1 to 4)

Query	Tempo	Contextuality	Redundancy	Q1	Q2	Q3	Q4	Q5
Janine1	6	4	5	No	No	No	No	Yes
Janine2	5	5	6	No	No	No	No	Yes
Janine3	5	5	6	No	No	No	No	Yes
Janine4	5	5	7	No	No	No	No	Yes
Ryan1	4	5	3	No	No	No	No	Yes
Ryan2	5	5	3	No	No	No	No	Yes
Ryan3	3	4	5	No	No	No	Yes	Yes
Ryan4	2	3	5	No	No	No	Yes	Yes
Stacey1	6	5	2	No	Yes	No	No	No
Stacey2	6	5	2	No	Yes	No	No	No
Stacey3	6	6	2	No	Yes	No	No	No
Stacey4	4	5	4	No	Yes	No	No	No

Surprisingly, the scores obtained for each run are very similar for the questions answering part of the evaluation. One exception concerns the character Ryan, for which one additional question is answered when choosing at least 15 shots. For all the runs, the redundancy score

Table 21: Questions used for qualitative evaluation

Character	Questions-nbr	Question
Janine	Q1	What is causing Ryan to be sick in bed?
Janine	Q2	How does Janine attempt to kill Ryan while in the hospital?
Janine	Q3	What happens when Janine attempts to play recording of Stacey?
Janine	Q4	Who stabbed Janine?
Janine	Q5	Who gives Janine the recording of Stacey?
Ryan	Q1	How does Janine attempt to kill Ryan in the hospital?
Ryan	Q2	What does Ryan do when Janine is lying in the hospital?
Ryan	Q3	Where is Ryan trapped?
Ryan	Q4	What does Ryan tell Phil he can do for him?
Ryan	Q5	Who is Ryan with when going to put his name on the babies birth cert?
Stacey	Q1	Who climbs up the roof to talk Stacey out of jumping off?
Stacey	Q2	What does Stacey reveal when in a cell with Janine, Kat, and Pat?
Stacey	Q3	What does Stacey admit to her mum in bedroom when mum is upset?
Stacey	Q4	Who confronts Stacey in restroom where Stacey finally admits to killing Archie?
Stacey	Q5	Who calls to Stacey’s door to tell her to get her stuff and go after Stacey’s mum had called the police?

improved with the number of shots included in the summary while the relation with the scores for tempo and contextuality seem to vary more. The scores are lower for the question answering evaluation part. This is rather unsurprising to us as we realized while deciding on a similarity measure score that it is challenging for humans to choose between two potentially interesting moments without knowing beforehand the questions included in the evaluation set. Overall, we consider that the results obtained speak in favour of using fan-made content as a starting point for such a task. As we did not try to optimize for tempo and contextuality, we believe there is some margin for improvement. However, the task of answering unknown questions remains an open challenge.

One of the key contributions of this work is to have demonstrated that despite some noise from face detection and recognition, this method enables to capture multiple important plot points for all three query characters. We also conclude that adding more shots to the summaries did, quite surprisingly, not always allow answering more key moments related questions. Finally, we would like to pinpoint the fact that the task of choosing important sequences that would answer unknown questions, is very challenging for humans. Indeed, when generating the runs, having read the summaries but not having watched the videos, we found it challenging to decide which sequences should be included in the summary. It would be interesting to know how much the score would improve if we would know the questions before evaluation.

4 Exploring the MeMAD Knowledge Graph

The MeMAD Knowledge Graph has been built and developed in the first two years of the project, with the goal of integrating all metadata from the various sources and providers into one common representation and infrastructure, thus facilitating the access and sharing of data across the project. In this section, we go through some of the final improvements made to the knowledge graph, as well as present the MeMAD Exploratory Search Engine, which provides a user interface for discovering and navigating through the content of the Knowledge Graph.

4.1 MeMAD ontology and controlled vocabularies

In line with the goal of unifying access to all data from the project, an effort of aligning descriptive tags in metadata that are common to all providers, namely *Genres*, *Roles* and *Languages* into controlled vocabularies has been made. A controlled vocabulary is usually a taxonomy or a classification scheme that covers all the possible values a metadata field can have, as well as the relationships among them.

In the first phase, we translated the vocabularies from INA and Yle into English (from French and Finnish respectively), thus building the MeMAD Ontology. Secondly, we match concepts from the MeMAD ontology from standard Classification Schemes such as the ones created by the European Broadcasting Union (which can be found at <https://www.ebu.ch/metadata/cs/>).

The resulting alignments are listed in tables 22 (INA) and 23 (Yle) for Genres (aligned with the EBU Content Genre Classification Scheme¹⁵), and tables 24 (INA) and 25 (Yle) for Roles (aligned with EBU Role Classification Scheme¹⁶). For all tables, we list the vocabulary used by INA and Yle respectively, and we introduce the MeMAD vocabulary word corresponding to it (we translate all terms into English with the help of domain experts). Finally, we attempt to align it with the EBU classification schemes to find either an exact match, a broad match (i.e. a concept that encompasses the one we have in the MeMAD corpus), or a close match i.e. concepts that are close semantically but not identical (for example “televised news” and “Daily news”). We note that language tags also received the same treatment, i.e. all language tags were translated into English.

Thanks to this vocabulary alignment, we can query the entire MeMAD corpus using the same (English) keywords.

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
	Adaptation	Adaptation			
	Animation	Animation			
	Bande annonce	Trailer	3.6.3.9	Trailer	exactMatch
	Best of	Best.of			
	Brève	Brief			
	Campagne d'information	Information campaign			
	Causerie	Chat	3.1.1.1.3	Chat	exactMatch
	Captation	Captation			
	Chronique	Chronicle			
	Conférence de presse	Press.conference			
	Court métrage	Short feature			
	Création audiovisuelle	Audiovisual creation			
	Création sonore	Sound.creation			
	Comédie de situation	Situational comedy			
	Cours d'enseignement	Course			
	Document à base d'archives	Archival document			
	Document amateur	Amateur document			
	Documentaire	Documentary	3.1.3.13	Documentary	exactMatch

Continued on next page

¹⁵https://www.ebu.ch/metadata/cs/ebu_ContentGenreCS_p.xml.htm

¹⁶https://www.ebu.ch/metadata/cs/ebu_RoleCodeCS_p.xml.htm

Table22 – continued from previous page					
Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
	Docuréalité	Docu-reality	3.1.7.1	Reality	closeMatch
	Docufiction	Docufiction			
	Dramatique	Drama	3.4	Fiction/Drama	exactMatch
	Débat	Debate	3.1.1.1.4	Debate	exactMatch
	Déclaration	Declaration			
	Emission à base de disques	Disc-based broadcast			
	Entretien	Interview	3.1.1.1.2	Interview	exactMatch
	Evocation scénarisée	Scripted evocation			
	Extrait	Extract			
	Feuilleton	Serial	3.4.2001	Popular drama	closeMatch
	Interlude	Interlude			
	Interprogrammes	Interprogrammes			
	Interprétation	Interpretation			
	Interview entretien	Interview	3.1.1.1.2	Interview	exactMatch
	Jeu	Game			
	Journal parlé	Spoken news	3.1.1.1	Daily news	closeMatch
	Journal télévisé	Televised news	3.1.1.1	Daily news	closeMatch
	Lecture	Reading			
	Libre antenne	Free airtime			
	Long métrage	Long feature			
	Magazine	Magazine	3.1.1.25	News magazine	closeMatch
	Making of	Making of			
	Message info	Info message			
	Message publicitaire	Publicity			
	Micro trottoir	Street interview			
	Mini programme	Mini programme			
	Musique savante	Art music			
	Plateau d'analyse	Studio analysis			
	Plateau en situation	Live set			
	Programme atypique	Atypical programming			
	Programme à base de clips	Clip-based programme			
	Oeuvre enregistrée en studio	Studio recording			
	Réalisation dans un lieu public	Public space production			
	Reality show	Reality show			
	Reconstitution	Reconstitution			
	Reportage	Report	3.1.1.3	Special Report	closeMatch
	Retransmission	Retransmission			
	Revue de presse	Press review			
	Récit portrait	Portrait story			
	Rétrospective	Retrospective			
	Sketch	Sketch			
	Spectacle TV	TV Spectable			
	Spectacle radio	Radio spectacle			
	Série	Series			
	Talk show	Talk show			
	Tout images	Il images			
	Tranche horaire	Time slot			
	Télécoaching	Telecoaching			
	Télé achat	Home shopping			
	Téléfilm	TV film	3.1.1.10.3	Film	closeMatch
	Télé réalité	Reality TV			
	Témoignage	Testimony			
	Vidéo clip	Video clip			
	Zapping	Zapping			

Table 22: Genre classification vocabulary and alignment for INA collection

The ontology is thus augmented by the vocabulary (as instances of `ebucore:Genre`, `ebucore:Role`, and `ebucore:Language`), and the list can be found at: <http://data.memad.eu/ontology>.

4.2 MeMAD automatically generated metadata

To integrate the newly generated metadata to the Knowledge Graph, we use the recently added class `ebucore:Annotation`. This class and its corresponding relation `ebucore:hasAnnotation` al-

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
Yle	Uutisbulletiini, uutislähetys	News bulletin	3.1.1	News/Pure information	exactMatch
	Makasiini	Magazine	3.1.1.25	News magazine	broadMatch
	Reportaasi, raportti	News report	3.1.1.3	Special Report	exactMatch
	Tapahtuma	Event	3.1.1.2	Special news/edition	closeMatch
	Lasten makasiiniohjelmat	Children's magazine			
	Muut lastenohjelmat	Other children's content			
	Ohjelmaesittelyt	Demonstrations, Trailer	3.6.3.9	Trailer	closeMatch
	Pelit	Games			
	Dokumentti	Documentary	3.1.3.13	Documentary	exactMatch
	Keskustelu, haastattelu	Interviews, discussions	3.1.1.1.2	Interview	exactMatch
	Lähetysvirta	Content feed			
	Asiaviihde	Factual entertainment	3.1.1.10.2	Entertainment	closeMatch
	Muut	Other	3.1.9.19.4	Other	exactMatch
	Urheilu-uutislähetys	Sports news bulletin	3.4.6.11	Sports	closeMatch
	Talk show	Talk show	3.1.1.1.3	Chat	closeMatch
	Asiareality	Factual reality	3.1.7.1	Reality	exactMatch
	Jumalanpalvelukset	Religious ceremony	3.1.9.19	Religious	closeMatch
	Muut hartausohjelmat	Other religious content	3.1.9.19	Religious	closeMatch
	taltiointi tai juonnettu	Concert	3.1.9.14	Concert/Live performance	exactMatch
	Juonnettu musiikkiohjelma	Hosted music show	3.6	Music	closeMatch
	Esitys (ooppera, baletti..)	Performance	3.1.9.14	Concert/Live performance	closeMatch
	Musiikkivideo	Music video			
	Musiikkikilpailut	Music competition			
	Muu musiikkiohjelma	Other music content	3.6	Music	exactMatch
	Toivekonsertti	Audience based concert	3.1.9.14	Concert/Live performance	closeMatch
	TV-elokuva	TV movie	3.1.1.10.3	Film	closeMatch
	Fiktiosarja	Fiction series	3.4	Fiction/Drama	exactMatch
	Animaatio, animaationsarja	Animation			
	Nukkenäytelmä, nukkesarja	Puppet play or series			
	(Elokuvateatteri)elokuva	Movie	3.1.1.10.3	Film	exactMatch
	Pistedraama, näytelmä	Drama / play	3.4	Fiction/Drama	closeMatch
	Kuunnelma	Radio drama	3.4	Fiction/Drama	broadMatch
	Luenta	Radio reading	3.1.1.10.5	Radio	broadMatch
	Tietokilpailut	Quiz show	3.5.2.1	Quiz	exactMatch
	Sketsiohjelmat (huumori, satiiri)	Humour	3.5.7.6	Humour	exactMatch
	Estradishow	Entertainment show	3.1.1.10.2	Entertainment	exactMatch
	Panel show	Panel show			
	Muut viihdeohjelmat	Other entertainment content	3.1.1.10.2	Entertainment	broadMatch
	Reality	Reality	3.1.7.1	Reality	exactMatch
	Kolumni	Feature (audio) article			closeMatch
	Podcast	Podcast	3.8.2.4	Podcasting	exactMatch
	Säätiedotus	Weather	3.1.1.13	Weather forecasts	exactMatch
	Ääniteos	Sonic art			
	Sarjadokumentti	Documentary series	3.1.3.13	Documentary	exactMatch
	Sekamuoto, asiaviihde	Mixed, factual entertainment			
	Keskustelu/Haastattelu/Debatti	Discussion	3.1.1.1.1	Discussion	exactMatch
	Tapahtumat	Events			
	Draamaohjelma	Drama	3.4	Fiction/Drama	exactMatch
	(Elokuvateatteri) elokuva	Cinematic film	3.1.1.10.3	Film	broadMatch
	Draama	Drama	3.4	Fiction/Drama	exactMatch
	Asiaohjelma	Factual	3.1	Non-Fiction/Informaion	closeMatch
	Asia	Factual	3.1	Non-Fiction/Informaion	closeMatch
	Musiikki	Music	3.6	Music	exactMatch

Table 23: Genre classification vocabulary and alignment for Yle collection

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
INA	"Auteur"	Author	22.2	Author/Screenplay/./Dramatiser	broadMatch
	Bruiteur"	Soundman	23.9	Foley Mixer/Sound Effect Person/Soundman	broadMatch
	Chef d'orchestre	Orchestrator	17.1.11	Orchestrator	exactMatch
	Commentateur	Commentator	25.21	Commentator	exactMatch
	Créateur des costumes	Costume Designer	28.1	Costume Designer/Illustrator	exactMatch
	Créateur des décors	Set Decorator	5.4.1	Set Decorator/Set Designer	exactMatch
	Dessinateur	Painter	5.6.6	Lead Painter	broadMatch
	Directeur de la photo	Cinematographer	6.2.1	Cinematographer	exactMatch
	Eclairagiste	Lighting Manager	4.28	Lighting/Shading Manager	closeMatch
	Interprète	Actor	25.9	Actor/Actress/Histrion/Thespian/Role Player	exactMatch
	Journaliste	Journalist	18.8	Broadcast Journalist/Video Journalist	closeMatch
	Journaliste reporter d'images	Photojournalist	18.9	Reporter	closeMatch
	Metteur en scène de théâtre"	Stage Designer	20.46	Stage Designer	closeMatch
	Mixage	Sound Mixer	11.22	Audio Editor/Sound Editor/./Sound Mixer	closeMatch
	Monteur	Editor	11.1	Editor/Visual Editor/./Video Editor	exactMatch
	Opérateur de prise de son	Sound Recordist	23.11	Sound Recordist / Sound Recorder	closeMatch
	Opérateur de prise de vue	Camera Operator	6.2.3	Camera Operator/Camera Person	closeMatch
	Participant	Participant	25.19	Participant	exactMatch
	Présentateur	Presenter	25.10	Anchor/Moderator/Presenter	exactMatch
	Producteur	Producer	10.1.2	Producer	exactMatch
	Réalisateur	Director	20.16	Director	exactMatch
	Rédacteur en chef	Editor in Chief	18.4	Editor in Chief	exactMatch
	Responsable d'édition	Editorial Coordinator	11.5	Editorial Coordinator	closeMatch
	Scripte	Script Supervisor	22.3	Script Supervisor/Continuity Person	closeMatch
	Traducteur	Translator	29.27	Translation/Translator	exactMatch
	Responsable d'édition	Editorial Coordinator	11.5	Editorial Coordinator	exactMatch

Table 24: Role classification vocabulary and alignment for INA collection

low us to seamlessly add the different results of automatic content analysis such as Face Recognition and Named Entity Recognition, content segmentation, and topic categorization. Following are some examples from each category.

```
# Content Segmentation
<annotation_part_1> rdf:type ebucore:Annotation;
    ebucore:annotationType memad:ContentSegmentation;
    ebucore:isAnnotationBy memad:EURECOM;
    ebucore:hasAnnotationBody <Automatic_Seg_1>;
    ebucore:hasAnnotationTarget <Programme_21>;
    ebucore:annotationConfidence "0.91"^^xsd:float.

# Face Recognition
<annotation_facerec_1> rdf:type ebucore:Annotation;
    ebucore:annotationType memad:FaceRecognition;
    ebucore:isAnnotationBy memad:EURECOM;
    ebucore:hasAnnotationBody wikidata:Q157;
    ebucore:hasAnnotationTarget <Programme_21#t=npt:120,121.3&xywh=percent:25,25,40,50>;
    ebucore:annotationConfidence "0.85"^^xsd:float.

# Topic Categoerization
<annotation_1> rdf:type ebucore:Annotation;
    ebucore:annotationType memad:TopicCategory;
    ebucore:isAnnotationBy memad:EURECOM;
    ebucore:hasAnnotationBody memad:Sport;
    ebucore:hasAnnotationTarget <Programme_21>;
    ebucore:annotationConfidence "0.88"^^xsd:float.

# Named Entity
memad:annotation_1 rdf:type ebucore:TextAnnotation;
    ebucore:annotationType memad:NamedEntity;
    ebucore:isAnnotationBy memad:EURECOM;
    ebucore:hasAnnotationBody wikidata:Q157;
```

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
Yle	Animaatiosuunnittelija	Animation Planner	4.8	Animation Supervisor	closeMatch
	Apulaisohjaaja	Assistant Director	20.17	First Assistant Director	closeMatch
	Arkistotoimittaja	Journalist, Archives			
	Asiantuntija	Expert	9.1	Expert	exactMatch
	Dramaturgi	Dramaturge	22.2	Author/Screenplay/.../Dramatiser	broadMatch
	Graafikko	Graphic Designer	5.9.1	Graphic Designer	exactMatch
	Graafinen suunnittelija	Graphic Designer	5.9.1	Graphic Designer	exactMatch
	Henkilöohjaaja	Director	10.1.1	Director	exactMatch
	Juontaja	Moderator	25.10	Anchor/Moderator/Presenter	closeMatch
	Järjestäjä	Archival Organizer			exactMatch
	Kirjailija	Writer	22.2	Author/Screenplay/.../Dramatiser	exactMatch
	Koreografi	Choreographer	25.17	Choreographer	exactMatch
	Kuvaussuunnittelija	Cinematographic Designer	6.2.19	Camera Supervisor	exactMatch
	Kuvaaja	Cinematographer	6.2.1	Cinematographer	exactMatch
	Kuvatoimittaja	Photo Editor	5.9.2	Graphic Editor	exactMatch
	Kuvaussihteeri	Script Supervisor	22.3	Script Supervisor / Continuity Person	exactMatch
	Käsitteittäjä	Scriptwriter	22.2	Author/Screenplay/.../Dramatiser	exactMatch
	Kääntäjä	Translator	29.27	Translation/Translator	exactMatch
	Lavastussuunnittelija	Stage Designer	20.46	Stage Designer	exactMatch
	Leikkaaja	Video Editor	11.1	Editor/Visual Editor/Film Editor/Video Editor	exactMatch
	Lukija (kertoja/speak)	Narrator	25.15	Narrator/Storyteller/Reader	broadMatch
	Meteorologi	Weather Forecaster			
	Musiikin suunnittelija	Music Supervisor	17.1.4	Music Supervisor/Coordinator	exactMatch
	Naamioitsija	Makeup Artist	13.2.2	Makeup Artist	exactMatch
	Näytelmäkirjailija	Playwright	22.5	Playwright	exactMatch
	Ohjaaja	Director TV/Radio	10.1.1	Director	broadMatch
	Pukusuunnittelija	Costume Designer	28.1	Costume Designer/Illustrator	exactMatch
	Puvustaja	Costumier	28.17	Costumer	exactMatch
	Selostaja	Commentator	25.21	Commentator	exactMatch
	Suunnittelija	Planner			
	Säveltäjä	Composer	17.1.7	Composer	exactMatch
	Taustatoimittaja	Researcher	20.22	Production Researcher	closeMatch
	Toimittaja	Journalist	18.8	Broadcast Journalist/Video Journalist	exactMatch
	Toimitussihteeri	Associate Editor	11.4	Assistant Editor/Assistant Visual Editor	closeMatch
	Tuotantopäällikkö	Productions Manager	20.10	Production Manager	exactMatch
	Tuottaja	Producer	20.1	Producer	exactMatch
	Uutispäällikkö	Editor in Chief, News	18.4	Editor in Chief	exactMatch
	Valokuvaaja	Photographer	6.4.1	Still Photographer	closeMatch
	Äänisuunnittelija	Sound Designer	11.24	Sound Designer/Supervising Sound Editor	exactMatch
	Äänittäjä	Sound Technician	23.10	Utility Sound Technician	closeMatch
	Tuotantokoordinaattori	Production Coordinator	20.14	Production Coordinator	exactMatch
	Toimituspäällikkö	Managing Editor			
	Lähetyskoordinaattori	Transmissions Coordinator			
	Sisältövastaava	Content Supervisor	22.3	Script Supervisor	closeMatch
	Päivätuottaja	Daily Producer	10.1.2	Producer	closeMatch

Table 25: Role classification vocabulary and alignment for Yle collection

```

ebucore:hasAnnotationTarget <TextLine_31>;
ebucore:characterStartIndex "8"^^xsd:int;
ebucore:characterEndIndex "13"^^xsd:int;
ebucore:annotationConfidence "0.92"^^xsd:float.

```

Listing 1: Examples of automatic annotations for the MeMAD Knowledge Graph

```

SELECT ?program ?title
WHERE
{
  ?program a ebucore:TVProgramme;
    ebucore:title ?title.
  ?annotation ebucore:hasAnnotationBody wikidata:Q157;
    ebucore:hasAnnotationTarget ?program.
}

```

Listing 2: Query all programs that either mention or visually show the French President *François Hollande* (wikidata QID = Q157)

4.3 MeMAD Exploratory Search Engine

The screenshot displays the MeMAD Exploratory Search Engine interface. At the top, there is a navigation bar with the MeMAD logo, the text 'Methods for Managing Audiovisual Data', and links for 'CHANNELS', 'COLLECTIONS', and 'PROGRAMMES'. A search bar contains the text 'Francois Hollande'. To the right of the search bar are links for 'ENGLISH' and 'PROFILE'. Below the navigation bar, the main content area shows '1322 search results'. On the left side, there is a sidebar with filters: 'Text search' (with a search bar), 'Genre' (with a dropdown menu), 'Theme' (with a dropdown menu), 'Language' (with a dropdown menu), and 'Options' (with checkboxes for 'With video' and 'With segments'). The main content area also includes a 'Sort by' dropdown menu and a 'Page 2' indicator. Below these, there is a grid of 8 video thumbnails. The first row contains four thumbnails: 'Rendez vous sur la Croisette: [émission du 22 mai 2014]', 'Sports: [émission du 19 mai 2014]', 'Sports: [émission du 26 mai 2014]', and 'Sports: [émission du 22 mai 2014]'. The second row contains four thumbnails: a woman speaking, a red abstract image, a modern building, and a red square with a white 'S' logo. At the bottom of the grid, there is a pagination bar with links for 'Previous', '1', '2' (highlighted), '3', '4', '5', '6', '...', '66', '67', and 'Next'.

Figure 22: The search page in KG Explorer

In order to provide the final user a good experience in accessing MeMAD resources and data, we developed KG Explorer, a fully-customisable web application which serves as an exploratory search engine for Knowledge Graphs. KG Explorer has the following features: a facet-based advanced search for programmes (Figure 22), channels and collections, a customised detail page for the main entities represented in the knowledge graph, the possibility for users to log in and to create personalized lists of favourites or saved items. The software can be

configured to adapt to different information domains, changing not only its aspect but also the queries for retrieving the data to display. KG Explorer is open source under Apache License 2.0 at <https://github.com/D2KLab/explorer>, while the application can be accessed at <https://explorer.memad.eu/>.

5 Conclusion

In this work package, we have first developed the so-called MeMAD Knowledge Graph which is available at <https://data.memad.eu/>. This knowledge graph is based on the standard EBU Core ontology as data model that we have ourselves contributing to. Hence, the latest version of this standard¹⁷ does reference MeMAD authors and as of today, MeMAD provides the largest implementation of this ontology. The knowledge graph makes also use of other popular ontologies (Web Annotations, PROV, NIF, etc.) and controlled vocabularies that we have aligned again with well-known references (e.g. IPTC NewsCodes, EBU TV Genres, etc.). In order to ease access to the knowledge graph, we have published an RESTful API. Finally, we have developed a prototype exploratory search engine named Explorer that enables to search and browse programs based on their original metadata as well as automatically generated metadata. This work has been entirely open sourced in the following github repositories:

- Converter to transform legacy metadata from CSV (INA) and XML (Yle) into RDF following the EBU Core ontology: <https://github.com/MeMAD-project/rdf-converter>
- MeMAD dynamic RESTful API: <https://github.com/MeMAD-project/api>
- MeMAD Explorer: <https://github.com/MeMAD-project/explorer>

One of the biggest research challenge of this work package was to develop novel methods that can identify the “micro-moments” of TV and Radio programs that would be important. We have decided to work on memorability as a proxy for assessing the importance of moments. We have participated for two years in the MediaEval Task on Predicting Media Memorability where we developed a multimodal approach combining visual, textual and visio-linguistic features which was ranked 1st (in 2019) and 3rd (in 2020). We have then evaluated the ability of these models to generalize on real-life programs of a very different length and genres (Sports Magazine and Lifestyle Magazine). We observe that this task is still an open research problem and that the models being produced do not generalize well. We also demonstrate the importance of having an adequate topical segmentation of the program. For this purpose, we developed two novel multimodal methods enabling to automatically generate a segmentation of a media. The first one is completely unsupervised, requiring solely to pre-define the number of expected segments, which can be based on previous episodes segmentation. The second one relies on existing content descriptions which provide another signal for performing a distant supervision of the content segmentation. Overall, these developments have also been open sourced in the following github repositories:

- Multimodal approaches for predicting media memorability as MediaEval 2019 and 2020: <https://github.com/MeMAD-project/media-memorability>
- Unsupervised approach for automatically segmenting a media: <https://github.com/MeMAD-project/content-segmentation>

Finally, this work package has largely aimed to propose enrichment services to attach to moments. We have considered three types of enrichment: topics, named entities and entire video summarization.

1. We proposed ToModAPI, a library enabling to dynamically perform training, inference, and evaluation for different topic modeling techniques. The RESTful API grant common interfaces and command for accessing the different models, make easier to compare them. A demo is available at <http://hyperted.eurecom.fr/topic>. We also compare numerous topic models and we empirically demonstrate that no one wins over all. Additional

¹⁷<https://www.ebu.ch/metadata/ontologies/ebucore/>

research work needs to be conducted to optimize topic models over multiple metrics including coherency and serendipity. Finally, we developed ZeSTE (Zero-Shot Topic Extraction) available at <https://zeste.tools.eurecom.fr/>, a novel method leveraging on the ConceptNet commonsense knowledge graph to predict the topics of a document while providing an explanation for this prediction. When applied to subtitles of medias, we empirically verify that we can reasonably re-predict the main topics of video sequences.

2. We have largely worked on named entity recognition and disambiguation and during this last year, we have proposed new approaches for tackling this problem. First, we proposed to cast the named entity recognition problem, which is always seen as a sequence labeling problem, as a graph classification problem and named our approach GraphNER. We have experimented with various graph embeddings approaches and we demonstrate that Graph Convolutional Network brings promising results, even if currently under the state of the art on the reference CoNLL 2003 dataset, but holding potential to inject more prior knowledge for this task, thus alleviating the need to have training corpora. We have also investigate how to extract named entity directly from the speech in an end-to-end fashion in a new approach named SpokenNER. Finally, we produced a new dataset available at https://drive.google.com/drive/u/0/folders/1NDW7GSN_ARBY19BQtNgdAKZmOg1ITtRv composed of ASR transcriptions and containing annotations about ASR errors and Named Entity Recognition and Named Entity Disambiguation against the Wikidata knowledge graph in the IOB2 format. The partners Aalto and Lingsoft have developed novel methods for disambiguating named entities in textual transcriptions using respectively deep learning and rule-based techniques. The rule-based approach developed by Lingsoft is being further industrialized in the context of the LSDISCO¹⁸ project funded by the European Language Grid (ELG) European project.
3. The final type of enrichment we have proposed correspond to an extractive video summarization of a program. In particular, we have proposed a novel multimodal method which is character centric, relying on our FaceRec library (<https://github.com/D2KLab/FaceRec>) and on fan-based comments available on fandom wikis. We have participated to the TRECVID Video Summarization task where our approach was ranked 1st by a large margin.

Overall, these developments have been open sourced in the following github repositories:

- Topic Modeling API (ToModAPI): <https://github.com/D2KLab/ToModAPI>
- Zero Shot Topic Extraction (ZeSTE): <https://github.com/D2KLab/ZeSTE>
- GraphNER: https://github.com/Siliam/graph_ner/
- SpokenNER: <https://github.com/Tetrix/E2E-NER-for-spoken-Finnish>
- Aalto Entity Linker:
<https://github.com/aalto-speech/Neural-Entity-Linking-for-Finnish>
- TRECVID Video Summarization 2020 approach: <https://github.com/MeMAD-project/trecvid-vsum>

¹⁸<https://live.european-language-grid.eu/catalogue/#/resource/projects/2204>

6 References

- [1] Alba García Seco de Herrera, Rukiye Savran Kiziltepe, Jon Chamberlain, Mihai Gabriel Constantin, Claire-Hélène Demarty, Faiyaz Doctor, Bogdan Ionescu, and Alan F. Smeaton. Overview of MediaEval 2020 predicting media memorability task: What makes a video memorable? In *Working Notes Proceedings of the MediaEval 2020 Workshop*, 2020.
- [2] Ismail Harrando, Benoit Huet, Alison Reboud, Raphaël Troncy, and Tiina Lindh-Knuutil. MeMAD Deliverable 3.2 - TV moments detection and linking. Technical report.
- [3] Alison Reboud, Ismail Harrando, Jorma Laaksonen, Danny Francis, Raphaël Troncy, and Héctor Laria Mantecón. Combining textual and visual modeling for predicting media memorability. In *MediaEval 2019: Multimedia Benchmark Workshop*, Sophia Antipolis, France, 2019.
- [4] David Azcona, Enric Moreu, Feiyan Hu, Tomás E Ward, and Alan F Smeaton. Predicting media memorability using ensemble models. In *MediaEval 2019: Multimedia Benchmark Workshop*, Sophia Antipolis, France, 2019.
- [5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.
- [6] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 776–780. IEEE, 2017.
- [7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA, 2016. IEEE.
- [9] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *International Conference on Computer Vision (ICCV)*, pages 4489–4497, Santiago, Chile, 2015. IEEE.
- [10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [12] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982—3992, Hong Kong, China, 2019. ACL.
- [13] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

- [14] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernamed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015. IEEE.
- [16] Lorin Sweeney, Graham Healy, and Alan F Smeaton. Leveraging audio gestalt to predict media memorability. In *Working Notes Proceedings of the MediaEval 2020 Workshop*, 2020.
- [17] Tony Zhao, Irving Fang, Jeffrey Kim, and Gerald Friedland. Multi-modal ensemble models for predicting video memorability. In *Working Notes Proceedings of the MediaEval 2020 Workshop*, 2020.
- [18] Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.
- [19] Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36, March 2002.
- [20] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine learning*, 34(1):177–210, 1999.
- [21] Martin Scaiano and Diana Inkpen. Getting more from segmentation evaluation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 362–366, Montréal, Canada, 2012. Association for Computational Linguistics.
- [22] Pasquale Lisena, Ismail Harrando, Oussama Kandakji, and Raphael Troncy. ToModAPI: A Topic Modeling API to Train, Use and Compare Topic Models. In *2nd International Workshop for Natural Language Processing Open Source Software (NLP-OSS)*, 2020.
- [23] Andrew Rosenberg and Julia Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, 2007.
- [24] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.
- [25] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 6442–6454. Association for Computational Linguistics, 2020.

- [26] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.
- [27] Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. End-to-end named entity recognition from english speech. *arXiv preprint arXiv:2005.11184*, 2020.
- [28] André Mansikkaniemi, Peter Smit, Mikko Kurimo, et al. Automatic construction of the finnish parliament speech corpus. In *INTERSPEECH*, volume 8, pages 3762–3766, 2017.
- [29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [30] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.
- [31] Martin Malmsten, Love Börjeson, and Chris Haffenden. Playing with words at the national library of sweden – making a swedish bert, 2020.
- [32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [33] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [34] Dejan Porjazovski, Juho Leinonen, and Mikko Kurimo. Named entity recognition for spoken finnish. In *Proceedings of the 2nd International Workshop on AI for Smart TV Content Production, Access and Delivery*, pages 25–29, 2020.
- [35] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.
- [36] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.
- [37] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *23rd international conference on Machine learning*, pages 369–376, 2006.
- [38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [39] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [40] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France, April 2012. Association for Computational Linguistics.
- [41] Lilia Perez Romero Michiel Hillebrand and Lynda Hardman. D3.8: Design guideline document for concept-based presentations. Technical report, March 2015.

- [42] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30. Association for Computational Linguistics, 2020.
- [43] Mika Härmäläinen. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345, 2019.
- [44] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA, 2020.
- [45] Mats Sjöberg, Hamed R. Tavakoli, Zhicun Xu, Héctor Laria Mantecón, and Jorma Laaksonen. PicSOM experiments in TRECVID 2018. In *22nd International Workshop on Video Retrieval Evaluation (TRECVID)*, Gaithersburg, USA, 2018.

A Dissemination activities

1. **Workshop organization** - 12/10/2020: AI4TV 2020: *2nd International Workshop on AI for Smart TV Content Production, Access and Delivery*, a workshop at ACM International Conference on Multimedia, Seattle, USA. Raphaël Troncy and Jorma Laaksonen chaired the workshop.
2. **Workshop presentation** - 19/11/2020: NLP-OSS 2020: *2nd International Workshop for Natural Language Processing Open Source Software*, a workshop at ACL EMNLP, Online. Ismail Harrando presented *ToModAPI: A Topic Modeling API to Train, Use and Compare Topic Models*.
3. **Workshop presentation** - 10/12/2020: TRECVID 2020: *TREC Video Retrieval Evaluation*, Online. Alison Reboud presented *Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization*.
4. **Workshop presentation** - 14/12/2020: MediaEval 2020: *MediaEval Benchmarking Initiative for Multimedia Evaluation*, Online. Alison Reboud and Ismail Harrando presented *Predicting Media Memorability with Audio, Video, and Text representations*.

B Appendices

B.1 EURECOM and AALTO's MediaEval 2020 workshop paper

This paper describes the approach that the EURECOM and AALTO teams published at the MediaEval 2020 Media Memorability Track where we ranked 3rd.

Predicting Media Memorability with Audio, Video, and Text representations

Alison Reboud^{*}, Ismail Harrando^{*}, Jorma Laaksonen⁺ and Raphaël Troncy^{*}

^{*}EURECOM, Sophia Antipolis, France

⁺Aalto University, Espoo, Finland

{alison.reboud, ismail.harrando, raphael.troncy}@eurecom.fr
jorma.laaksonen@aalto.fi

ABSTRACT

This paper describes a multimodal approach proposed by the MeMAD team for the MediaEval 2020 “Predicting Media Memorability” task. Our best approach is a weighted average method combining predictions made separately from visual, audio, textual and visiolinguistic representations of videos. Our best model achieves Spearman scores of 0.101 and 0.078, respectively, for the short and long term predictions tasks.

1 INTRODUCTION

Considering video memorability as a useful tool for digital content retrieval as well as for sorting and recommending an ever growing number of videos, the Predicting Media Memorability task aims at fostering the research in the field by asking its participants to automatically predict both a short and a long term memorability score for a given set of annotated videos. The full description for this task is provided in [5]. Last year’s best approaches for both the long term [10] and short term tasks [2] rely on multimodal features. Our method is inspired from last year’s best approaches but also acknowledges the specifics of the 2020’s edition dataset. More specifically, because in comparison to last year’s set of videos, the TRECVID videos contain more actions, our model uses video features and image features for multiple frames. In addition, because this year sound was included in the videos, our model includes audio features. Finally, a key contribution of our approach is to test the relevance of visiolinguistic representation for the Media Memorability task. Our final model¹ is a multimodal weighted average with visual and audio deep features extracted from the videos, textual features from the provided captions and visiolinguistic features.

2 APPROACH

We trained separate models for the short and long term predictions using originally a 6-fold cross-validation of the training set, which means that we typically had 492 samples for training and 98 samples for testing each model.

¹<https://github.com/MeMAD-project/media-memorability>

2.1 Audio-Visual Approach

Our audio-visual memorability prediction scores are based on using a feed-forward neural network with a concatenation of video and audio features in the input, one hidden layer of units and one unit in the output layer. The best performance was obtained with 2575-dimensional features consisting of the concatenation of 2048-dimensional I3D [3] video features and 527-dimensional audio features. Our audio features encode the occurrence probabilities of the 527 classes of the Google AudioSet Ontology [6] in each video clip. The hidden layer uses ReLU activations and dropout during the training phase, while the output unit is sigmoidal. The training of the network used the Adam optimizer. The features, the number of training epochs and the number of units in the hidden layer were selected with the 6-fold cross-validation. For short term memorability prediction, the optimal number of epochs was 750 and the optimal hidden layer size 80 units, whereas for the long term prediction these figures were 260 and 160, respectively.

We also experimented with other types of features and their combinations. These include the ResNet [7] features extracted just from the middle frames of the clips as this approach worked very well last year. The contents of this year’s videos are, however, such that genuine video features I3D and C3D [13] work better than still image features. When I3D and AudioSet features are used, C3D features do not bring any additional advantage.

2.2 Textual Approach

Our textual approach leverages the video descriptions provided by the organizers. First, all the provided descriptions are concatenated by video identifier to get one string per video. To generate the textual representation of the video content, we used the following methods:

- Computing TF-IDF, removing rare (less than 4 occurrences) and stopwords and accounting for frequent 2-grams.
- Averaging GloVe embeddings for all non-stopwords words using the pre-trained 300d version [9].
- Averaging BERT [4] token representations (keeping all the words in the descriptions up to 250 words per sentence).
- Using Sentence-BERT [11] sentence representations. We use the distilled version that is fine-tuned for the STS Textual Similarity Benchmark².

For each representation, we experimented with multiple regression models and finetuned the hyper-parameters for each model

²<https://huggingface.co/sentence-transformers/distilbert-base-nli-stsb-mean-tokens>

using the 6-fold cross-validation on the training set. For our submission, we used the *Averaging GloVe embeddings* with a Support Machine Regressor with an RBF kernel and a regulation parameter $C = 1e - 5$.

We also attempted enhancing the provided descriptions with additional captions automatically generated using the DeepCaption³ software. We did not see an improvement in the results, which is probably due to the nature of the clips provided for this year's edition (as DeepCaption is trained on static stock images from MS COCO and TGIF datasets).

2.3 Visiolinguistic Approach

ViLBERT [8] is a task-agnostic extension of BERT that aims to learn the associations and links between visual and linguistic properties of a concept. It has a two-stream architecture, first modelling each modality (i.e. visual and textual) separately, and then fusing them through a set of attention-based interactions (co-attention). ViLBERT is pre-trained using the Conceptual Captions data set (3.3M image-caption pairs) [12] on masked multi modal learning and multi-modal alignment prediction. We used a frozen pre-trained model which was fine-tuned twice, first on the task of Video-Question Answering (VQA) [1] and then on the 2019 MediaEval Memorability task and dataset.

The 1024-dimensional features extracted for the two modalities can be combined in different ways. In our experiment, multiplying textual and visual feature vectors performed the best for short term memorability prediction but using the sole visual feature vectors worked better for long term memorability prediction. Averaging the features extracted from 6 frames performed better than only using only the middle frame. We experimented with the same set of regression models as for the textual approach. In our submission, we used a Support Machine Regressor with a regulation parameter $C = 1e - 5$ and an RBF or Poly kernel respectively for short and long term scores prediction.

3 RESULTS AND ANALYSIS

We have prepared 5 different runs following the task description defined as follows:

- run1 = Audio-Visual Score
- run2 = Visiolinguistic Score
- run3 = Textual Score
- run4 = $0.5 * \text{run1} + 0.2 * \text{run2} + 0.3 * \text{run3}$
- run5 = run4 with LT scores for LT task

For the Long Term task, all models except *run5* use exclusively short-term scores. For runs 4 and 5, we normalise the scores obtained from runs 1, 2 and 3 before combining them.

Table 1 provides the Spearman score obtained for each run when performing a 6-folds cross-validation on the training set. We observe that our models use only the training set, as the annotations on the later-provided development set did not yield better results. We hypothesize that this is due to the fewer number of annotations per video available as many videos had a score for 1, for instance, which we do not observe on the training set.

³<https://github.com/aalto-cbir/DeepCaption>

Table 1: Average Spearman score obtained on a 6-folds cross validation of the Training set

Method	Short Term	Long Term
run1	0.2899	0.179
run2	0.214	0.1309
run3	0.2506	0.1372
run4	0.3104	0.2038
run5	0.067	0.1700

Table 2: Results on the Test set for Short Term (ST) and Long Term (LT) memorability

Method	SpearmanST	PearsonST	SpearmanLT	PearsonLT
run1	0.099	0.09	0.077	0.0855
run2	0.098	0.085	-0.017	0.011
run3	0.073	0.091	0.019	0.049
run4	0.101	0.09	0.078	0.085
run5	0.101	0.09	0.067	0.066
AvgTeams	0.058	0.066	0.036	0.043

We present in Table 2 the final results obtained on the test set using models trained on the full training set composed of 590 videos. We observe that the weighted average method which uses short term scores works the best for both short and long term prediction, obtaining results which are approximately double the mean Spearman score obtained across the teams. Our best results (Spearman scores) on the test set are however significantly worse than the ones we obtained on average over the 6-folds of the training set suggesting that the test set is quite different from the training set. The results for Long Term prediction are always worse than the ones for Short Term prediction. Finally, both our scores and the mean score across team are below the ones obtained for the 2018 and 2019 videos.

4 DISCUSSION AND OUTLOOK

This paper describes a multimodal weighted average method proposed for the 2020 Predicting Media Memorability task of MediaEval. One of the key contribution of this paper is to have shown that based on our experiments during the model construction or testing phase, in comparison to image, audio and text, video features performed the best. Similarly to last year, short term scores predictions correlated better with long term scores than the predictions made when training directly on long term scores. Finally considering the difference of results obtained between the training and test set, it would be interesting to investigate further the differences between these datasets in terms of content (video, audio and text) and annotation. We conclude that generalizing this type of task to different video genres and characteristics remain a scientific challenge.

Acknowledgements

This work has been partially supported by the European Union's Horizon 2020 research and innovation programme via the project MeMAD (GA 780069).

REFERENCES

- [1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, Santiago, Chile.
- [2] David Azcona, Enric Moreu, Feiyan Hu, Tomás E Ward, and Alan F Smeaton. 2019. Predicting media memorability using ensemble models. In *MediaEval 2019: Multimedia Benchmark Workshop*. Sophia Antipolis, France.
- [3] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4724–4733.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. ACL, Minneapolis, Minnesota, USA, 4171–4186.
- [5] Alba García Seco de Herrera, Rukiye Savran Kiziltepe, Jon Chamberlain, Mihai Gabriel Constantin, Claire-Hélène Demarty, Faiyaz Doctor, Bogdan Ionescu, and Alan F. Smeaton. 2020. Overview of MediaEval 2020 Predicting Media Memorability task: What Makes a Video Memorable?. In *Working Notes Proceedings of the MediaEval 2020 Workshop*.
- [6] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, Louisiana, USA, 776–780.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, Nevada, USA, 770–778.
- [8] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *33rd Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada.
- [9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, Melbourne, Australia, 1532–1543.
- [10] Alison Reboud, Ismail Harrando, Jorma Laaksonen, Danny Francis, Raphaël Troncy, and Héctor Laria Mantecón. 2019. Combining Textual and Visual Modeling for Predicting Media Memorability. In *MediaEval 2019: Multimedia Benchmark Workshop*. Sophia Antipolis, France.
- [11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, Hong Kong, China, 3982–3992.
- [12] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL, Melbourne, Australia, 2556–2565.
- [13] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *International Conference on Computer Vision (ICCV)*. IEEE, Santiago, Chile, 4489–4497.

B.2 EURECOM and AALTO's TRECVID VSUM 2020 workshop paper

This paper describes the approach that the EURECOM and AALTO teams published at the TRECVID 2020 Video Summarization Task where we ranked 1st.

Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization

ISMAIL HARRANDO*, ALISON REBOUD*, PASQUALE LISENA, and RAPHAËL TRONCY, EURECOM, France
JORMA LAAKSONEN, ANJA VIRKKUNEN, and MIKKO KURIMO, Aalto University, Finland

This paper describes a fan-driven and character-centered approach proposed by the MeMAD team for the 2020 TRECVID [Awad et al. 2020] Video Summarization Task. Our approach relies on fan-made content and, more precisely, on the BBC EastEnders episode synopses from its Fandom Wiki¹. This additional data source is used together with the provided videos, scripts and master shot boundaries. We also use BBC EastEnders characters' images crawled from the Google search engine in order to train a face recognition system. All our runs use the same method, but with varying constraints regarding the number of shots and the maximum duration of the summary. The shots included in the summaries are the ones whose transcripts and visual content have the highest similarity with sentences from the synopsis. The runs submitted are as follows:

- MeMAD1: 5 shots with highest similarity scores and the total duration of the summary is < 150 sec;
- MeMAD2: 10 shots with highest similarity scores and the total duration of the summary is < 300 sec;
- MeMAD3: 15 shots with highest similarity scores and the total duration of the summary is < 450 sec;
- MeMAD4: 20 shots with highest similarity scores and the total duration of the summary is < 600 sec.

Surprisingly, the scores obtained for each run are very similar for the questions answering part of the evaluation. One exception concerns the character Ryan, for which one additional question is answered when choosing at least 15 shots. For all the runs, the redundancy score improved with the number of shots included in the summary while the relation with the scores for tempo and contextuality seem to vary more. The scores are lower for the question answering evaluation part. This is rather unsurprising to us as we realized while deciding on a similarity measure score that it is challenging for humans to choose between two potentially interesting moments without knowing beforehand the questions included in the evaluation set. Overall, we consider that the results obtained speak in favour of using fan-made content as a starting point for such a task. As we did not try to optimize for tempo and contextuality, we believe there is some margin for improvement. However, the task of answering unknown questions remains an open challenge.

ACM Reference Format:

Ismail Harrando, Alison Reboud, Pasquale Lisena, Raphaël Troncy, Jorma Laaksonen, Anja Virkkunen, and Mikko Kurimo. 2021. Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization. In *Proceedings of TRECVID 2020, International Workshop on Video*

*Both authors contributed equally to this research.

¹https://eastenders.fandom.com/wiki/EastEnders_Wiki

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

TRECVID 2020, December 8–11, 2020, Virtual Conference

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Retrieval Evaluation (TRECVID 2020). ACM, New York, NY, USA, 3 pages.
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Considering video summarization as an important task for digital content retrieval and reuse, the TRECVID [Awad et al. 2020] Video Summarization Task (VSUM) 2020 aims at fostering the research in the field by asking its participants to automatically summarize “the major life events of specific characters over a number of weeks of programming on the BBC EastEnders TV series”². More precisely, for three different characters of the series, the participants have to submit 4 summaries with respectively 5, 10, 15 and 20 automatically selected shots. These generated summaries are evaluated by the assessors according to their tempo, contextuality and redundancy as well as with regards to how well they contain answers to a set of questions unknown to the participants before submission. In addition to the videos, the episodes transcripts are provided by the organizers.

We propose a character centered content summary approach based on fan-written synopses. The approach relies on scraping the Fandom EastEnders Wiki content for the episode synopsis and casting, in order to align them with the corresponding episodes. We include the shots that obtain the best similarity score with a sentence from the synopsis in our runs.

2 APPROACH

Our fan-driven and character centered approach is presented in Figure 1.

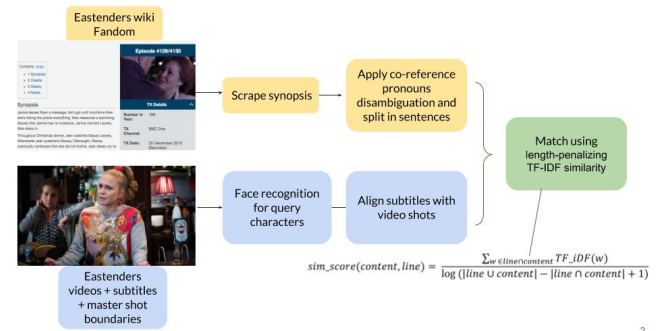


Fig. 1. Fan-driven and character centered approach

²<https://www-nlpir.nist.gov/projects/tv2020/vsum.html>

2.1 Scraping Synopses From the Fandom Wiki and Selecting Shots

The first step of our approach consists in scraping synopses available on the Fandom EastEnders Wiki³.

Our main hypothesis is that every sentence (ending with a period) represents an important event to be added to the final video summary. We scrape the Synopsis and the Cast sections for each episode broadcasted between the dates of the provided episodes. The mapping between the episodes and their dates is in `eastenders.collection.xml` provided by the challenge organizers.

In parallel, we extract the shots in which the three characters of interest appear from the video. We run the Face Celebrity Recognition library⁴, a system that relies on pictures crawled from search engines using the actor’s name as search keyword. In our experiments, we have added "EastEnders" to the character names in order to avoid retrieving pictures of different people with the same name. For each picture, faces are detected using the MTCNN algorithm and the FaceNet model is applied to obtain face embeddings. Following the assumption that the majority of faces are actually representing the searched actor, other faces – e.g. person portrayed together with the actor – are automatically filtered out by removing outliers until the cosine similarity of face embeddings has a standard deviation below a threshold of 0.24 which has been empirically defined.

The remaining faces are used to train a multi-class SVM classifier, which is used to label the faces detected on frames. For more consistent results between frames, the Simple Online and Realtime Tracking algorithm (SORT) has been included, returning groups of detection of the same person in consecutive frames.

We select the shots displaying any of the the three characters of interests, keeping only those detection having a confidence score greater than 0.5. We also tried to use speaker diarisation to corroborate the visual information about the characters. However, given the limitations of the current technologies in terms of number of characters and the difficulty of identifying the character corresponding to each voice, we could not pursue the idea further.

2.2 Synopses and Transcript Pre-Processing

A synopsis for each episode was created using the provided files *eastenders.collection.xml* and *eastenders.episodeDescriptions.xml*. Since these were "EastEnders Omnibus" episodes, they correspond to multiple actual weekday episodes. We use the dates and the continuation to generate one synopsis for each "long" episode (typically made of 4 episodes). We then split the synopses into sentences and performed coreference resolution on the synopses to explicit character mentions using <https://github.com/huggingface/neuralcoref>. In parallel, the provided XML transcripts were also converted into timestamped text and aligned with the given shot segmentation. Finally, both the synopses sentences and shot transcripts were lower cased, stop words removed and lemmatized.

We also produced automatically-generated visual captions following the method presented by the PicSOM Group of Aalto University’s submissions for the TRECVID2018 VTT task [Sjöberg et al. 2018]. The hypothesis is that by describing the visual information of a

³<https://eastenders.fandom.com/wiki/EastEndersWiki>

⁴<https://github.com/D2KLab/Face-Celebrity-Recognition>

Table 1. Average score for each run and team

TeamRun	Percentage
MeMAD1	31%
MeMAD2	31%
MeMAD3	35%
MeMAD4	32%
NIIUIT1	9%
NIIUIT2	8%
NIIUIT3	8%
NIIUIT4	6%

shot, visual captions could complement well the dialog transcript and therefore allow for a better matching between the shots and synopses sentences.

2.3 Matching and Runs Generation

We perform a synopsis sentence / shot transcript pairwise comparison by generating a similarity score. We define similarity between two sentences as the sum of TF-IDF weights (computed on the transcript) for each word appearing in both of them, divided by the log length of the concatenation of both sentences, thus penalizing long sentences that match with many transcript lines.

Next, we order the shot by similarity score, picking only the best match for each shot (but not the other way around). This gives us scenes we are sure to appear in the summary, but not necessarily any guarantee about how important these scenes are. We also performed the pairwise comparison adding the automatically generated captions. A qualitative assessment revealed, however, that the captions were too noisy to complement well the transcript. We also make sure that if a line of dialog runs through the next shot, we include the next shot as well to improve the smoothness of the viewing. However, this heuristics was only relevant for the longest run (20 shots). Each run is made by selecting the N most matching shots out of the top, in chronological order.

3 RESULTS AND ANALYSIS

The final results for the two teams which have participated in TRECVID VSUM are presented in Table 1 while the detailed scores of our approach are presented in Table 2. Our method obtains the best overall score for each of the 4 required runs. The mean scores (range 1 - 7. High is best) for tempo, contextuality and redundancy are all above average (respectively 4.75, 4.75, 4.1) despite the fact that our method does not specifically attempt to optimise these metrics. However, in terms of question answering, the results show that the shots selected did not allow to answer more than two (at best) of the five questions. More specifically, Table 3 shows (in bold) the questions that were answered in at least one of our runs. We notice that most of the questions started either with 'What' or 'Who' and that our approach performed equally for both types of questions.

4 DISCUSSION AND OUTLOOK

This paper describes a character centered video summarization method based on fan-made content, subtitles and face recognition.

Table 2. Detailed score for MeMAD’s approach

Query	Tempo	Contextuality	Redundancy	Q1	Q2	Q3	Q4	Q5
Janine1	6	4	5	No	No	No	No	Yes
Janine2	5	5	6	No	No	No	No	Yes
Janine3	5	5	6	No	No	No	No	Yes
Janine4	5	5	7	No	No	No	No	Yes
Ryan1	4	5	3	No	No	No	No	Yes
Ryan2	5	5	3	No	No	No	No	Yes
Ryan3	3	4	5	No	No	No	Yes	Yes
Ryan4	2	3	5	No	No	No	Yes	Yes
Stacey1	6	5	2	No	Yes	No	No	No
Stacey2	6	5	2	No	Yes	No	No	No
Stacey3	6	6	2	No	Yes	No	No	No
Stacey4	4	5	4	No	Yes	No	No	No

Table 3. Questions used for qualitative evaluation

Character	Questions-nbr	Question
Janine	Q1	What is causing Ryan to be sick in bed?
Janine	Q2	How does Janine attempt to kill Ryan while in the hospital?
Janine	Q3	What happens when Janine attempts to play recording of Stacey?
Janine	Q4	Who stabbed Janine?
Janine	Q5	Who gives Janine the recording of Stacey?
Ryan	Q1	How does Janine attempt to kill Ryan in the hospital?
Ryan	Q2	What does Ryan do when Janine is lying in the hospital?
Ryan	Q3	Where is Ryan trapped?
Ryan	Q4	What does Ryan tell Phil he can do for him?
Ryan	Q5	Who is Ryan with when going to put his name on the babies birth cert?
Stacey	Q1	Who climbs up the roof to talk Stacey out of jumping off?
Stacey	Q2	What does Stacey reveal when in a cell with Janine, Kat, and Pat?
Stacey	Q3	What does Stacey admit to her mum in bedroom when mum is upset?
Stacey	Q4	Who confronts Stacey in restroom where Stacey finally admits to killing Archie?
Stacey	Q5	Who calls to Stacey’s door to tell her to get her stuff and go after Stacey’s mum had called the police?

One of the key contribution of this paper is to have demonstrated that despite some noise from face detection and recognition, this method enables to capture multiple important plot points for all three query characters. We also conclude that adding more shots to the summaries did, quite surprisingly, not always allow to answer more key moments related questions. Finally, we would like to pinpoint the fact that the task of choosing important sequences that would answer unknown questions, is very challenging for humans. Indeed, when generating the runs, having read the summaries but not having watched the videos, we find it challenging to decide which sequences should be included in the summary. It would be interesting to know how much the score would improve if we would know the questions before evaluation.

REFERENCES

- George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. 2020. TRECVID 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of the TRECVID 2020 Workshop*. NIST, Gaithersburg, MD, USA.
- Mats Sjöberg, Hamed R. Tavakoli, Zhicun Xu, Héctor Laria Mantecón, and Jorma Laaksonen. 2018. PicSOM Experiments in TRECVID 2018. In *Proceedings of the TRECVID 2018 Workshop*. NIST, Gaithersburg, MD, USA.

B.3 EURECOM's NLP-OSS 2020 workshop paper

This paper describes the ToModAPI API developed by EURECOM and published at the NLP-OSS 2020 Workshop colocated with EMNLP 2020.

ToModAPI: A Topic Modeling API to Train, Use and Compare Topic Models

Pasquale Lisena, Ismail Harrando, Oussama Kandakji and Raphaël Troncy

EURECOM, Sophia Antipolis, France
{firstname.lastname}@eurecom.fr

Abstract

From LDA to neural models, different topic modeling approaches have been proposed in the literature. However, their suitability and performance is not easy to compare, particularly when the algorithms are being used in the wild on heterogeneous datasets. In this paper, we introduce ToModAPI (*TOpic MOdeling API*), a wrapper library to easily train, evaluate and infer using different topic modeling algorithms through a unified interface. The library is extensible and can be used in Python environments or through a Web API.

1 Introduction

The analysis of massive volumes of text is an extremely expensive activity when it relies on not-scalable manual approaches or crowdsourcing strategies. Relevant tasks typically include textual document classification, document clustering, keywords and named entities extraction, language or sequence modeling, etc. In the literature, topic modeling and topic extraction, which enable to automatically recognise the main subject (or topic) in a text, have attracted a lot of interest. The predicted topics can be used for clustering documents, for improving named entity extraction (Newman et al., 2006), and for automatic recommendation of related documents (Luostarinen and Kohonen, 2013).

Several topic modeling algorithms have been proposed. However, we argue that it is hard to compare and to choose the most appropriate one given a particular goal. Furthermore, the algorithms are often evaluated on different datasets and different scoring metrics are used. In this work, we have selected some of the most popular topic modeling algorithms from the state of the art in order to integrate them in a common platform, which homogenises the interface methods and the evaluation

metrics. The result is ToModAPI¹ which allows to dynamically train, evaluate, perform inference on different models, and extract information from these models as well, making it possible to compare them using different metrics.

The remaining of this paper is organised as follows. In Section 2, we describe some related works and we detail some state-of-the-art topic modeling techniques. In Section 3, we provide an overview of the evaluation metrics usually used. We introduce ToModAPI in Section 4. We then describe some datasets (Section 5) that are used in training to perform a comparison of the topic models (Section 6). Finally, we give some conclusions and outline future work in Section 7.

2 Related Work

Aside from a few exceptions (Blei and McAuliffe, 2007), most topic modeling works propose or apply unsupervised methods. Instead of learning the mapping to a pre-defined set of topics (or labels), the goal of these methods consists in assigning training documents to N unknown topics, where N is a required parameter. Usually, these models compute two distributions: a Document-Topic distribution which represents the probability of each document to belong to each topic, and a Topic-Word distribution which represents the probability of each topic to be represented by each word present in the documents. These distributions are used to predict (or infer) the topic of unseen documents.

Latent Dirichlet Allocation (LDA) is an unsupervised statistical modeling approach (Blei et al., 2003) that considers each document as a *bag of words* and creates a randomly assigned document-topic and word-topic distribution. Iterating over words in each document, the distributions are updated according to the probability that a document

¹ToModAPI: TOpic MOdeling API

or a word belongs to a certain topic. The **Hierarchical Dirichlet Process (HDP)** model (Teh et al., 2006) is another statistical approach for clustering grouped data such as text documents. It considers each document as a group of words belonging with a certain probability to one or multiple components of a mixture model, i.e. the topics. Both the probability measure for each document (distribution over the topics) and the base probability measure – which allows the sharing of clusters across documents – are drawn from Dirichlet Processes (Ferguson, 1973). Differently from many other topic models, HDP infers the number of topics automatically.

Gibbs Sampling for a DMM (GSDMM) applies the Dirichlet Multinomial Mixture model for short text clustering (Yin and Wang, 2014). This algorithm works computing iteratively the probability that a document join a specific one of the N available clusters. This probability consist in two parts: 1) a part that promotes the clusters with more documents; 2) a part that advantages the movement of a document towards similar clusters, i.e. which contains a similar word-set. Those two parts are controlled by the parameters α and β . The simplicity of GSDMM provides a fast convergence after some iterations. This algorithm consider the given number of clusters given as an upper bound and it might end up with a lower number of topics. From another perspective, it is somehow able to infer the optimal number of topics, given the upper bound.

Pre-trained Word vectors such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) can help to enhance topic-word representations, as achieved by the **Latent Feature Topic Models (LFTM)** (Nguyen et al., 2015). One of the LFTM algorithms is *Latent Feature LDA (LF-LDA)*, which extends the original LDA algorithm by enriching the topic-word distribution with a latent feature component composed of pre-trained word vectors. In the same vein, the **Paragraph Vector Topic Model (PVTM)** (Lenz and Winker, 2020) uses doc2vec (Le and Mikolov, 2014) to generate document-level representations in a common embedding space. Then, it fits a Gaussian Mixture Model to cluster all the similar documents into a predetermined number of topics – i.e. the number of GMM components.

Topic modeling can also be performed via linear-algebraic methods. Starting from the the high-

dimensional term-document matrix, multiple approaches can be used to lower its dimensions. Then, we consider every dimension in the lower-rank matrix as a latent topic. A straightforward application of this principle is the **Latent Semantic Indexing model (LSI)** (Deerwester et al., 1990), which uses Singular Value Decomposition as a means to approximate the term-document matrix (potentially mediated by TF-IDF) into one with less rows – each one representing a latent semantic dimension in the data – and preserving the similarity structure among columns (terms). **Non-negative Matrix Factorisation (NMF)** (Paatero and Tapper, 1994) exploits the fact that the term-document matrix is non-negative, thus producing not only a denser representation of the term-document distribution through the matrix factorisation but guaranteeing that the membership of a document to each topic is represented by a positive coefficient.

In recent years, neural network approaches for topic modeling have gained popularity giving birth to a family of **Neural Topic Models (NTM)** (Cao et al., 2015). Among those, **doc2topic (D2T)**² uses a neural network which separately computes N -dimensional embedding vectors for words and documents – with N equal to the number of topics, before computing the final output using a sigmoid activation. The distributions topic-word and document-topic are obtained by getting the final weights on the two embedding layers. Another neural topic model, the **Contextualized Topic Model (CTM)** (Bianchi et al., 2020) uses Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) – a neural transformer language model designed to compute sentences representations efficiently – to generate a fixed-size embedding for each document to contextualise the usual Bag of Words representation. CTM enhances the *Neural-ProdLDA* (Srivastava and Sutton, 2017) architecture with this contextual representation to significantly improve the coherence of the generated topics.

Previous works have tried to compare different topic models. A review of statistical topic modeling techniques is included in Newman et al. (2006). A comparison and evaluation of LDA and NMF using the coherence metric is proposed by O’Callaghan et al. (2015). Among the libraries for performing topic modeling, *Gensim* is undoubtedly the most known one, providing implementations of

²<https://github.com/sronnqvist/doc2topic>

several tools for the NLP field (Řehůřek and Sojka, 2010). Focusing on topic modeling for short texts, *STMM* includes 11 different topic models, which can be trained and evaluated through command line (Qiang et al., 2019). The *Topic Modelling Open Source Tool*³ exposes a web graphical user interface for training and evaluating topic models, LDA being the only representative so far. The *Promoss Topic Modelling Toolbox*⁴ provides a unified Java command line interface for computing a topic model distribution using LDA or the *Hierarchical Multi-Dirichlet Process Topic Model (HMDP)* (Kling, 2016). However, it does not allow to apply the computed model on unseen documents.

3 Metrics

The evaluation of machine learning techniques often relies on accuracy scores computed comparing predicted results against a ground truth. In the case of unsupervised techniques like topic modeling, the ground truth is not always available. For this reason, in the literature, we can find:

- metrics which enable to evaluate a topic model independently from a ground truth, among which, coherence measures are the most popular ones for topic modeling (Röder et al., 2015; O’Callaghan et al., 2015; Qiang et al., 2019);
- metrics that measure the quality of a model’s predictions by comparing its resulting clusters against ground truth labels, in this case a topic label for each document.

3.1 Coherence metrics

The coherence metrics rely on the joint probability $P(w_i, w_j)$ of two words w_i and w_j that is computed by counting the number of documents in which those words occur together divided by the total number of documents in the corpus. The documents are fragmented using sliding windows of a given length, and the probability is given by the number of fragments including both w_i and w_j divided by the total number of fragments. This probability can be expressed through the *Pointwise Mutual Information (PMI)*, defined as:

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)} \quad (1)$$

³<https://github.com/opeyemibami/Topic-Modelling-Open-Source-Tool>

⁴<https://github.com/gesiscss/promoss>

A small value is chosen for ϵ , in order to avoid computing the logarithm of 0. Different metrics based on PMI have been introduced in the literature, differing in the strategies applied for token segmentation, probability estimation, confirmation measure, and aggregation. The **UCI coherence** (Röder et al., 2015) averages the PMI computed between pairs of topics, according to:

$$C_{UCI} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N PMI(w_i, w_j) \quad (2)$$

The **UMASS coherence** (Röder et al., 2015) relies instead on a differently computed joint probability:

$$C_{UMASS} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \quad (3)$$

The **Normalized Pointwise Mutual Information (NPMI)** (Chiarcos et al., 2009) applies the PMI in a confirmation measure for defining the association between two words:

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log(P(w_i, w_j) + \epsilon)} \quad (4)$$

NPMI values go from -1 (never co-occurring words) to +1 (always co-occurring), while the value of 0 suggests complete independence. This measure can be applied also to word sets. This is made possible using a vector representation in which each feature consists in the NPMI computed between w_i and a word in the corpus W , according to the formula:

$$\vec{v}(w_i) = \left\{ NPMI(w_i, w_j) | w_j \in W \right\} \quad (5)$$

In ToModAPI, we include the following four metrics⁵:

- C_{NPMI} applies NPMI as in Eqn (4) to couples of words, computing their joint probabilities using sliding windows;
- C_V compute the cosine similarity of the vectors – as defined in Eqn (5) – related to each word of the topic. The NPMI is computed on sliding windows;
- C_{UCI} as in Eqn (2);
- C_{UMASS} as in Eqn (3).

⁵We use the implementation of these metrics as provided in Gensim. The window size is kept at the default values.

Additionally, we include a **Word Embeddings-based Coherence** as introduced by Fang et al. (2016). This metric relies on pre-trained word embeddings such as GloVe or word2vec and evaluate the topic quality using a similarity metric between its top words. In other words, a high mutual embedding similarity between a model’s top words reflects its underlying semantic coherence. In the context of this paper, we will use the sum of mutual cosine similarity computed on the Glove vectors⁶ of the top $N = 10$ words of each topic:

$$C_{WE} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \cos(v_i, v_j) \quad (6)$$

where v_i and v_j are the GloVe vectors of the words w_i and w_j .

All metrics aggregate the different values at topic level using the arithmetic mean, in order to provide a coherence value for the whole model.

3.2 Metrics which relies on a ground truth

The most used metric that relies on a ground truth is the **Purity**, defined as the fraction of documents in each cluster with a correct prediction (Hajjem and Latiri, 2017). A prediction is considered correct if the original label coincides with the original label of the majority of documents falling in the same topic prediction. Given L the set of original labels and T the set of predictions:

$$Purity(T, L) = \frac{1}{|T|} \sum_{i \in T} \max_{j \in L} |T_j \cap L_j| \quad (7)$$

In addition, we include in the API the following metrics used in the literature for evaluating the quality of classification or clustering algorithms, applied to the topic modeling task:

1. **Homogeneity**: a topic model output is considered homogeneous if all documents assigned to each topic belong to the same ground-truth label (Rosenberg and Hirschberg, 2007);
2. **Completeness**: a topic model output is considered complete if all documents from one ground-truth label fall into the same topic (Rosenberg and Hirschberg, 2007);
3. **V-Measure**: the harmonic mean of Homogeneity and Completeness. A V-Measure of

1.0 corresponds to a perfect alignment between topic model outputs and ground truth labels (Rosenberg and Hirschberg, 2007);

4. **Normalized Mutual Information (NMI)** is the ratio between the mutual information between two distributions – in our case, the prediction set and the ground truth – normalised through an aggregation of those distributions’ entropies (Lancichinetti et al., 2009). The aggregation can be realised by selecting the minimum/maximum or applying the geometric/arithmetic mean. In the case of arithmetic mean, NMI is equivalent to the V-Measure.

For these metrics, we use the implementations provided by scikit-learn (Pedregosa et al., 2011).

4 ToModAPI: a Topic Modeling API

We now introduce ToModAPI, a Python library which harmonises the interfaces of topic modeling algorithms. So far, 9 topic modeling algorithms have been integrated in the library (Table 1).

For each algorithm, the following interface methods are exposed:

- `train` which requires in input the path of a dataset and an algorithm-specific set of training parameters;
- `topics` which returns the list of trained topics and, for each of them, the 10 most representative words. Where available, the weights of those words in representing the topic are given;
- `topic` which returns the information (representative words and weights) about a single topic;
- `predict` which performs the topic inference on a given (unseen) text;
- `get_training_predictions` which provides the final predictions made on the training corpus. Where possible, this method is not performing a new inference on the text, but returns the predictions obtained during the training;
- `coherence` which computes the chosen coherence metric – among the ones described in Section 3.1 – on a given dataset;
- `evaluate` which evaluate the model predictions against a given ground truth, using the metrics described in Section 3.2.

⁶We use a Glove model pre-trained on Wikipedia 2014 + Gigaword 5, available at <https://nlp.stanford.edu/projects/glove/>

Algorithm	Acronym	Source implementation
Latent Dirichlet Allocation	LDA	http://mallet.cs.umass.edu/ (McCallum, 2002) (JAVA)
Latent Feature Topic Models	LFTM	https://github.com/datquocnguyen/LFTM (JAVA)
Doc2Topic	D2T	https://github.com/sronqvist/doc2topic
Gibbs Sampling for a DMM	GSDMM	https://github.com/rwalk/gsdmm
Non-Negative Matrix Factorization	NMF	https://radimrehurek.com/gensim/models/nmf.html
Hierarchical Dirichlet Processing	HDP	https://radimrehurek.com/gensim/models/hdpmodel.html
Latent Semantic Indexing	LSI	https://radimrehurek.com/gensim/models/lsmi.html
Paragraph Vector Topic Model	PVTM	https://github.com/davidlenz/pvtm
Context Topic Model	CTM	https://github.com/MilaNLPProc/contextualized-topic-models

Table 1: Algorithms included in ToModAPI, with their source implementation. The original implementation of those model is in Python unless specified otherwise.

The structure of the library, which relies on class inheritance, is easy to extend with the addition of new models. In addition to allowing the import in any Python environment and use the library offline, it provides the possibility of automatically build a web API, in order to access to the different methods through HTTP calls. Table 2 provides a comparison between the ToModAPI, Gensim and STMM. Given that we wrap some Gensim models and methods (i.e. for coherence computation), some similarities between it and our work can be observed.

The software is distributed under an open source license⁷. A demo of the web API is available at <http://hyperted.eurecom.fr/topic>.

5 Datasets and pre-trained models

Together with the library, we provide pre-trained models trained on two different datasets having different characteristics (20NG and AFP). A common pre-processing is performed on the datasets before training, consisting of:

- Removing numbers, which, in general, do not contribute to the broad semantics;
- Removing the punctuation and lower-casing;
- Removing the standard English stop words;
- Lemmatisation using Wordnet, in order to deal with inflected forms as a single semantic item;
- Ignoring words with 2 letters or less. In facts, they are mainly residuals from removing punctuation – e.g. stripping punctuation from *people's* produces *people* and *s*.

The same pre-processing is also applied to the text before topic prediction.

⁷<https://github.com/D2KLab/ToModAPI>

5.1 20 NewsGroups

The 20 NewsGroups collection (20NG) (Lang, 1995) is a popular dataset used for text classification and clustering. It is composed of English news documents, distributed fairly equally across 20 different categories according to the subject of the text. We use a reduced version of this dataset⁸, which excludes all the documents composed by the sole header while preserving an even partition over the 20 categories. This reduced dataset contains 11,314 documents. We pre-process the dataset in order to remove irrelevant metadata – consisting of email addresses and news feed identifiers – keeping just the textual content. The average number of words per document is 142.

5.2 Agence France Presse

The Agence France Presse (AFP) publishes daily up to 2000 news articles in 5 different languages⁹, together with some metadata represented in the NewsML XML-based format. Each document is categorised using one or more subject codes, taken from the IPTC NewsCode Concept vocabulary¹⁰. In case of multiple subjects, they are ordered by relevance. In this work, we only consider the first level of the hierarchy of the IPTC subject codes. We extracted a dataset containing 125,516 news documents in English and corresponding to the production of AFP for the year 2019, with 237 words per document on average.

Table 3 summarizes the number of documents for each topic in those two datasets. In AFP, a single document can be assigned to multiple subject, so we take each assignment into account. The two

⁸<https://github.com/selva86/datasets/>

⁹The catalogue can be explored at <http://medialab.afp.com/afp4w/>

¹⁰<http://cv.iptc.org/newscodes/subjectcode/>

library	Gensim	STMM	ToModAPI
algorithms	8: LDA, LDA Sequence, LDA multicore, NMF, LSI, HDP, Author-topic model, DTM	11: LDA, LFTM, DMM, BTM, WNTM, PTM, SATM, ETM, GPU-DMM, GPU-PDMM, LF-DMM	9: LDA, LFTM, D2T, GSDMM, NMF, HDP, LSI, PVTM, CTM
language	Python	Java	Python
focus	general	short text	general
training	✓	✓	✓
inference	✓	✓	✓
corpus predictions	(by inferencing the corpus)	✓	✓
coherence metrics	C_{umass} , C_v , C_{uci} , C_{npmi}	C_{umass}	C_{umass} , C_v , C_{uci} , C_{npmi}
Evaluation with Ground Truth	-	purity, NMI	purity, homogeneity, completeness, v-measure, NMI
usage	import in script	command line	import in script, web API

Table 2: Comparison between topic modeling libraries. For details about the acronyms, refer to the documentation

datasets present multiple differences: total number of documents, distribution of documents per subject, and the fact that for AFP, one document can have multiple subjects.

20NG		AFP	
rec.sport.hockey	600	Politics	47277
soc.religion.christian	599	Sport	36901
rec.motorcycles	598	Economy, Business, Finance	31042
rec.sport.baseball	597	Unrest, Conflicts and War	21140
sci.crypt	595	Crime, Law and Justice	16977
sci.med	594	Art, Culture, Entertainment	8586
rec.autos	594	Social Issues	7609
comp.windows.x	593	Disasters and Accidents	5893
sci.space	593	Human Interest	4159
comp.os.ms-windows.misc	591	Environmental Issue	4036
sci.electronics	591	Science and Technology	3502
comp.sys.ibm.pc.hardware	590	Religion and Belief	3081
misc.forsale	585	Lifestyle and Leisure	3044
comp.graphics	584	Labour	2570
comp.sys.mac.hardware	578	Health	2535
talk.politics.mideast	564	Weather	1159
talk.politics.guns	546	Education	734
alt.atheism	480		
talk.politics.misc	465		
talk.religion.misc	377		
Total	11314	Total	125516

Table 3: Number of documents per subject in 20NG (20 topics) and AFP (17 topics)

5.3 Wikipedia Corpus

We also describe the Wikipedia corpus (Wiki)¹¹, which is a readily extracted and organised snapshot from 2013 that includes pages with at least 20 page views in English. This corpus has been used in other works, for example, for computing word embeddings (Leimeister and Wilson, 2018). The corpus is distributed with some pre-processing already applied, like lower-casing and punctuation

¹¹https://storage.googleapis.com/lateral-datadumps/wikipedia_utf8_filtered_20pageviews.csv.gz

stripping. However, we performed additional operations such as lemmatisation, stop-word and small word (2 characters or less) removal. The dataset consists of around 463k documents with 498M words. This corpus will not be used for training but only for evaluating the models (trained on 20NG or AFP) in order to reflect on the generalisation of the topics models.

6 Experiment and Results

We empirically evaluate the performances of the topic modeling algorithms described in Section 2 on the two datasets presented in Section 5 using the metrics detailed in Section 3. For each algorithm, we trained two different models, respectively on 20NG and AFP corpus. The number of topics – when required by the algorithm – has been set to 20 and 7 when training on 20NG and AFP, respectively, in order to mimic the original division in class labels of the corpora (except for GSDMM and HDP which infer the optimal number of topics). Each model trained on either 20NG or AFP is tested against the same dataset and the Wikipedia dataset to compute each metric.

Table 4 shows the average coherence scores of the topics computed on the 20NG dataset, together with the standard deviation, while the results of Table 5 refer to models computed on the AFP dataset. The results differ depending on the studied metric and the evaluation dataset. LFTM generalises better when evaluated against the Wikipedia corpus, probably thanks to the usage of pre-trained word vectors on large corpora. Overall, LDA has the best results on all metrics, always being among

	C_v				C_{NPMI}				C_{UMASS}				C_{UCI}			
	20NG		wiki		20NG		wiki		20NG		wiki		20NG		wiki	
CTM	0.56	(0.15)	0.46	(0.24)	-0.04	(0.19)	-0.06	(0.16)	-5.78	(5.27)	-4.28	(3.94)	-3.09	(4.18)	-2.51	(3.95)
D2T	0.57	(0.14)	0.51	(0.10)	0.01	(0.11)	0.05	(0.05)	-2.94	(1.67)	-2.02	(0.49)	-1.56	(2.39)	0.16	(0.81)
GSDMM	0.50	(0.18)	0.41	(0.20)	0.00	(0.19)	-0.04	(0.09)	-3.86	(2.88)	-2.45	(1.04)	-2.02	(3.16)	-1.44	(2.26)
HDP	0.44	(0.21)	0.48	(0.24)	-0.09	(0.17)	-0.04	(0.10)	-5.59	(5.04)	-3.25	(3.18)	-5.59	(5.04)	-2.21	(2.64)
LDA	0.64	(0.14)	0.55	(0.16)	0.10	(0.08)	0.07	(0.06)	-1.98	(0.68)	-1.75	(0.45)	0.27	(1.30)	0.53	(0.88)
LFTM	0.53	(0.09)	0.56	(0.17)	-0.01	(0.10)	0.07	(0.06)	-2.97	(3.15)	-1.72	(0.69)	-1.47	(2.47)	0.58	(0.76)
LSI	0.53	(0.22)	0.41	(0.11)	0.03	(0.16)	-0.04	(0.10)	-3.25	(2.16)	-2.64	(1.08)	-1.37	(2.89)	-1.69	(2.59)
NMF	0.61	(0.19)	0.52	(0.15)	0.10	(0.15)	-0.02	(0.12)	-2.37	(1.61)	-3.08	(4.83)	-0.03	(2.24)	-1.27	(2.97)
PVTM	0.54	(0.09)	0.46	(0.11)	0.06	(0.04)	0.04	(0.06)	-1.63	(0.82)	-1.52	(0.54)	0.21	(0.92)	0.25	(0.74)

Table 4: The mean and standard deviation of different coherence metrics computed on 2 reference corpora 20NG and Wikipedia. The models have been trained on 20NG.

	C_v				C_{NPMI}				C_{UMASS}				C_{UCI}			
	AFP		wiki		AFP		wiki		AFP		wiki		AFP		wiki	
CTM	0.54	(0.15)	0.56	(0.28)	-0.05	(0.17)	-0.04	(0.09)	-6.56	(5.94)	-3.47	(2.96)	-2.75	(3.73)	-1.49	(2.17)
D2T	0.58	(0.14)	0.45	(0.10)	0.06	(0.07)	-0.01	(0.07)	-2.25	(0.49)	-2.44	(0.73)	-0.02	(0.93)	-1.07	(1.42)
GSDMM	0.51	(0.12)	0.58	(0.17)	0.09	(0.07)	0.03	(0.11)	-1.72	(0.47)	-2.73	(1.31)	0.70	(0.66)	-0.29	(1.59)
HDP	0.42	(0.10)	0.69	(0.22)	0.02	(0.07)	0.01	(0.16)	-2.23	(0.92)	-2.74	(2.63)	-0.20	(1.05)	-0.63	(2.86)
LDA	0.65	(0.10)	0.54	(0.11)	0.11	(0.04)	0.06	(0.06)	-1.40	(0.23)	-1.88	(0.48)	0.80	(0.30)	0.25	(0.89)
LFTM	0.59	(0.14)	0.54	(0.20)	0.06	(0.10)	0.06	(0.12)	-1.97	(2.40)	-1.91	(2.19)	0.11	(2.08)	0.22	(2.58)
LSI	0.58	(0.12)	0.55	(0.14)	0.07	(0.09)	0.05	(0.11)	-1.80	(0.47)	-2.59	(1.37)	0.09	(0.96)	-0.36	(1.87)
NMF	0.67	(0.12)	0.46	(0.12)	0.13	(0.06)	0.04	(0.07)	-1.27	(0.29)	-1.73	(0.69)	0.95	(0.42)	0.07	(1.26)
PVTM	0.52	(0.12)	0.51	(0.09)	0.07	(0.06)	0.04	(0.04)	-1.16	(0.34)	-1.56	0.86	0.49	(0.41)	0.14	(0.63)

Table 5: The mean and standard deviation of different coherence metrics computed on 2 reference corpora AFP and Wikipedia. The models have been trained on AFP.

the top ones in terms of coherence. When trained on AFP, all topic models benefit of a bigger dataset; this results in generally higher scores and in different algorithms maximising specific metrics.

We also consider the time taken by the different techniques for different tasks like training and getting prediction (Table 6). The results have been collected selecting the best of 3 different calls. The inference time has been computed using the models trained on the 20NG dataset, on a small sentence of 18 words¹². The table shows LDA leading in training, while the longest execution time belongs to LFTM. The inference time for all models is in the order of few seconds or even less than 1 for GSDMM, HDP, LSI and PVTM. The manipulation of BERT embeddings makes CTM inference more time-consuming. The inference timing for D2T is not computed because its implementation is not available yet.

7 Conclusions and Future Work

In this paper, we introduced ToModAPI, a library and a Web API to easily train, test and evaluate topic models. 9 algorithms are already included in the library, while new ones will be added in future. Other evaluation metrics for topic modeling

have been proposed (Wallach et al., 2009) and will be included in the API for enabling a complete evaluation. Among these, metrics based on word embeddings are gaining particular attention (Ding et al., 2018). For further exploiting the advantage of having a common interface, we will study ways to automatically tune each model’s hyper-parameters such as the right number of topics, find an appropriate label for the computed topics, optimise and use the models in real world applications. Finally, future work includes a deeper comparison of the models trained on different datasets.

	Training		Inference
	20NG	AFP	
CTM	544	9,262	19
D2T	192	5,892	-
GSDMM	1,194	21,881	0
HDP	430	7,020	0
LDA	80	1,334	2
LFTM	3,119	15,100	1
LSI	383	6,716	0
NMF	357	6,320	5
PVTM	193	3,757	0

Table 6: Model comparison from a time (in seconds) delay standpoint for training and inference.

¹²“Climate change is a global environmental issue that is affecting the lands, the oceans, the animals, and humans”

Acknowledgments

This work has been partially supported by the French National Research Agency (ANR) within the ASRAEL (grant number ANR-15-CE23-0018) and ANTRACT (grant number ANR-17-CE38-0010) projects, and by the European Union's Horizon 2020 research and innovation program within the MeMAD (grant agreement No. 780069) and SILKNOW (grant agreement No. 769504) projects.

References

- Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2020. [Pre-training is a hot topic: Contextualized document embeddings improve topic coherence](#). ArXiv.
- David M. Blei and Jon D. McAuliffe. 2007. Supervised Topic Models. In *20th International Conference on Neural Information Processing Systems (NIPS)*, pages 121–128.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A Novel Neural Topic Model and Its Supervised Extension. In *AAAI Conference on Artificial Intelligence*.
- Christian Chiarcos, Richard Eckart de Castilho, and Manfred Stede. 2009. *Von der Form zur Bedeutung: Texte automatisch verarbeiten - From Form to Meaning: Processing Texts Automatically*. Narr Francke Attempto Verlag GmbH + Co. KG.
- Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.
- Ran Ding, Ramesh Nallapati, and Bing Xiang. 2018. Coherence-Aware Neural Topic Modeling. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 830–836, Brussels, Belgium.
- Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. 2016. Using Word Embedding to Evaluate the Coherence of Topics from Twitter Data. In *39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1057–1060.
- Thomas S. Ferguson. 1973. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230.
- Malek Hajjem and Chiraz Latiri. 2017. Combining IR and LDA Topic Modeling for Filtering Microblogs. In *21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES)*, pages 761–770, Marseille, France.
- Christoph Kling. 2016. *Probabilistic models for context in social media*. doctoral thesis, Universität Koblenz-Landau, Universitätsbibliothek.
- Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3).
- Ken Lang. 1995. NewsWeeder: Learning to Filter News. In *20th International Conference on Machine Learning (ICML)*, pages 331–339.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *31st International Conference on Machine Learning (ICML)*, pages 1188–1196, Beijing, China.
- Matthias Leimeister and Benjamin J. Wilson. 2018. [Skip-gram word embeddings in hyperbolic space](#). Arxiv.
- David Lenz and Peter Winker. 2020. Measuring the diffusion of innovations with paragraph vector topic models. *PLOS ONE*, 15:1–18.
- Tapio Luostarinen and Oskar Kohonen. 2013. Using Topic Models in Content-Based News Recommender Systems. In *19th Nordic Conference of Computational Linguistics (NODALIDA)*.
- Andrew Kachites McCallum. 2002. [MALLET: A Machine Learning for Language Toolkit](#).
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *26th International Conference on Neural Information Processing Systems (NIPS)*, volume 2, pages 3111–3119, Lake Tahoe, NV, USA.
- David Newman, Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2006. Analyzing Entities and Topics in News Articles Using Statistical Topic Models. In *Intelligence and Security Informatics*, pages 93–104.
- Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving Topic Models with Latent Feature Word Representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.
- Derek O’Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. 2015. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657.
- Pentti Paatero and Unto Tapper. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. 2019. [Short Text Topic Modeling Techniques, Applications, and Performance: A Survey](#). Arxiv.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *LREC Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992, Hong Kong, China.
- Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *8th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 399–408.
- Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic.
- Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *International Conference on Learning Representations (ICLR)*.
- Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.
- Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *26th Annual International Conference on Machine Learning (ICML)*, pages 1105–1112.
- Jianhua Yin and Jianyong Wang. 2014. A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering. In *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 233–242.

B.4 EURECOM's ACL 2021 submission

This paper describes a thorough comparison of topic models made by EURECOM and submitted at ACL 2021.

Abstract

From probabilistic to neural models, different topic modelling algorithms have been proposed in the literature. However, their performance is not easy to evaluate, especially when these algorithms are used on heterogeneous datasets. In this paper, we present several approaches to topic modelling, an overview of the different metrics used to compare their performance, and the challenges of conducting such a comparison. We empirically evaluate the performance of 9 topic models from the literature on different settings reflecting a variety of real-life situations in term of dataset size, number of topics, and distribution of topics, using both metrics that rely only on the intrinsic characteristics of the results (coherence), as well as the agreement between the resulting topic distribution and their ground truth. Our findings reveal some shortcomings regarding the common practices in topic models evaluation.

1 Introduction

The analysis of massive volumes of text is an extremely expensive activity when it relies on manual approaches or crowdsourcing strategies which are non-scalable. Relevant tasks typically include textual document classification, document clustering, keywords and named entities extraction, language or sequence modelling, etc. In the literature, topic modelling and topic extraction, which enable to automatically recognise the main subject of a text, has attracted a lot of interest. The predicted topics can be used for clustering documents, for improving named entity extraction (Newman et al., 2006), and for recommendation of related document.

Several topic modelling algorithms have been proposed. However, we argue that it is hard to make any fair comparison among the different approaches. They are often evaluated on different

datasets (or sometimes, subsets of datasets) and different scoring metrics are used.

In this work, we selected some of the most used topic modelling algorithms from the literature and we provide a thorough comparison using a common evaluation protocol, where each topic model is evaluated on several datasets and using a variety of metrics that range from intrinsic evaluation of the clustering quality to ones that assess the alignment between the extracted topics and the human-assigned labels. We analyse the results and we discuss about the differences in performances across the different algorithms, datasets and parameters.

The remaining of this paper is organised as follows. In Section 2, we describe some related work, detailing some state-of-the-art topic modelling techniques. Different metrics for evaluating topic models are introduced in Section 3, while Section 4 describes the datasets we use for this purpose. In Section 5, we extensively analyse 9 topic models using coherence and ground truth accuracy. Finally, we provide some conclusions in Section 6.

2 Related Work

2.1 Topic Modelling Techniques

Except few exceptions (Blei and McAuliffe, 2007), most topic modelling approaches rely on unsupervised training methods. Instead of learning the mapping to a pre-defined set of topics (or labels), the goal of these methods consists in assigning text documents from a collection to one of N topics, where N is a required parameter. Typically, these models compute two distributions: a Document-Topic distribution which represents the probability of each document to belong to each topic, and a Topic-Word distribution which represents the probability of each topic to be represented by each word present in the documents. These distributions are used to predict the topic of unseen documents.

One of the first yet still widely used techniques is **Latent Dirichlet Allocation (LDA)** (Blei et al., 2003), an unsupervised statistical modelling approach that considers each document as a *bag of words* and creates a randomly assigned document-topic and word-topic distribution. Iterating over words in each document, the distributions are updated according to the probability that a document or a word belongs to a certain topic. The **Hierarchical Dirichlet Process (HDP)** model (Teh et al., 2006) considers instead each document as a group of words belonging with a certain probability to one or multiple components of a mixture model, i.e. the topics. Both the probability measure for each document (distribution over the topics) and the base probability measure – which allows the sharing of clusters across documents – are drawn from Dirichlet Processes (Ferguson, 1973). Unlike most other topic models, HDP infers the number of topics automatically.

Gibbs Sampling for a DMM (GSDMM) applies the Dirichlet Multinomial Mixture model for short text clustering (Yin and Wang, 2014). This algorithm works computing iteratively the probability that a document join a specific one of the N available clusters. This probability consist in two parts: 1) a part that promotes the clusters with more documents; 2) a part that advantages the movement of a document towards similar clusters, i.e. which contains a similar word-set. Those two parts are controlled by the parameters α and β . The simplicity of GSDMM provides a fast convergence after some iterations. This algorithm consider the given number of clusters given as an upper bound and it might end up with a lower number of topics. From another perspective, it is somehow able to infer the optimal number of topics, given the upper bound.

Recently, pre-trained Word vectors such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) have been used to help to enhance topic-word representations, as achieved by the **Latent Feature Topic Models (LFTM)** (Nguyen et al., 2015). One of the LFTM algorithms is *Latent Feature LDA (LF-LDA)*, which extends the original LDA algorithm by enriching the topic-word distribution with a latent feature component composed of pre-trained word vectors. In the same vein, the **Paragraph Vector Topic Model (PVTM)** (Lenz and Winker, 2020) uses doc2vec (Le and Mikolov, 2014) to generate document-level representations in a common em-

bedding space. Then, it fits a Gaussian Mixture Model to cluster all the similar documents into a predetermined number of topics – i.e. the number of GMM components.

Topic modelling can also be performed via linear-algebraic methods. Starting from the the high-dimensional term-document matrix, multiple approaches can be used to lower its dimensions. Then, we consider every dimension in the lower-rank matrix as a latent topic. A straightforward application of this principle is the **Latent Semantic Indexing model (LSI)** (Deerwester et al., 1990), which uses Singular Value Decomposition as a means to approximate the term-document matrix (potentially mediated by TF-IDF) into one with less rows – each one representing a latent semantic dimension in the data – and preserving the similarity structure among columns (terms). **Non-negative Matrix Factorisation (NMF)** (Paatero and Tapper, 1994) exploits the fact that the term-document matrix is non-negative, thus producing not only a denser representation of the term-document distribution through the matrix factorisation but guaranteeing that the membership of a document to each topic is represented by a positive coefficient.

In recent years, neural network approaches for topic modelling have gained popularity giving birth to a family of **Neural Topic Models (NTM)** (Cao et al., 2015). Among those, **doc2topic (D2T)**¹ uses a neural network which separately computes N -dimensional embedding vectors for words and documents (with N = number of topics) before computing the final output using a sigmoid activation. The distributions topic-word and document-topic are obtained by getting the final weights on the two embedding layers. The **Contextualized Topic Model (CTM)** (Bianchi et al., 2020) uses Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) – a neural transformer language model designed to compute sentences representations efficiently – to generate a fixed-size embedding for each document to contextualise the usual Bag of Words representation. CTM enhances the *Neural-ProdLDA* (Srivastava and Sutton, 2017) architecture with this contextual representation to significantly improve the coherence of the generated topics.

¹<https://github.com/sronnqvist/doc2topic>

2.2 Topic Models Comparison

While no extensive comparison of recent topic models has been made, some previous works have tried to compare different topic models on certain datasets and metrics. A review of statistical topic modelling techniques is included in Newman et al. (2006). A comparison and evaluation of LDA and NMF on some news and tweets corpora using multiple coherence metrics is proposed by O’Callaghan et al. (2015). Schofield and Mimno (2016) provide a comparison resulting from the effect of pre-processing on the performance of LDA on multiple corpora. Jelodar et al. (2017) offers a survey of topic modelling techniques based on LDA, as well as their different applications in recent literature. Yi and Allan (2009) and Alexander and Gleicher (2016) offer a comparison between several topic models evaluated as tools for performing Information Retrieval downstream tasks such as *Topic Alignment*, *Change Comparison*, *Document Retrieval* and *Query Expansion*. Several evaluation metrics based on top-words analysis was suggested by Newman et al. (2010a). Alghamdi and Alfalqi (2015) compare 4 topic models (LDA, LSI, PLSA and CTM): this survey studied both their capability in modelling static topics, as well as in detecting topic change over time, highlighting the strengths and weaknesses of each. Newman et al. (2010b) set out to automatically assess the coherence and interpretability of the topics learned by topic models using an evaluation protocol involving human subjects, showing that top-words PMI (computed from different textual sources) has a high correlation with human scores for a wide variety of topics. Burkhardt and Kramer (2019) provide a survey for the adjacent task of multi-label topic models, underlining its challenges and promising directions. Finally, Qiang et al. (2020) give an extensive performance evaluation of multiple topic models in the context of the *Short Text Topic modelling* sub-task (e.g. tweets), showing how the traditional topic models that rely on word co-occurrence statistics do not fare as well self-aggregation and DMM-based methods, while also providing a comparison on the computational efficiency of said models.

2.3 Metrics

While our work utilises multiple comparison metrics (detailed in Section 3.1), it is worth highlighting that many other evaluation metrics were proposed in the literature to expose different charac-

teristics of the studied topic models such as Classification Accuracy and Perplexity (Qiang et al., 2020), Entropy and Held-out Likelihood (Schofield and Mimno, 2016), Stability (Alexander and Gleicher, 2016), and Top-word Ranking (Greene et al., 2014), whereas finding a universally useful metric for topic modelling evaluation is still an open problem (Blei, 2012).

3 Metrics

The evaluation of machine learning techniques often relies on accuracy scores computed comparing predicted results against a ground truth. In the case of unsupervised techniques like topic modelling, the ground truth is not always available. For this reason, in the literature, we can find:

- metrics which enable to evaluate a topic model independently from a ground truth, among which, coherence measures are the most popular ones (Röder et al., 2015; O’Callaghan et al., 2015; Qiang et al., 2020);
- metrics that measure the quality of a model’s predictions by comparing its resulting clusters against ground truth labels, in this case a topic label for each document.

3.1 Coherence Metrics

The coherence metrics rely on the joint probability $P(w_i, w_j)$ of two words w_i and w_j that is computed by counting the number of documents in which those words occur together divided by the total number of documents in the corpus. The documents are fragmented using sliding windows of a given length, and the probability is given by the number of fragments including both w_i and w_j divided by the total number of fragments. This probability can be expressed through the *Pointwise Mutual Information (PMI)*, defined as:

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)} \quad (1)$$

A small value is chosen for ϵ , in order to avoid computing the logarithm of 0. Different metrics based on PMI have been introduced in the literature, differing in the strategies applied for token segmentation, probability estimation, confirmation measure, and aggregation. The **UCI coherence** (Röder et al., 2015) averages the PMI computed between pairs of topics, according to:

$$C_{UCI} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N PMI(w_i, w_j) \quad (2)$$

The **UMASS coherence** (Röder et al., 2015) relies instead on a different joint probability:

$$C_{UMASS} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \quad (3)$$

The **Normalized Pointwise Mutual Information (NPMI)** (Chiarcos et al., 2009) applies the PMI in a confirmation measure for defining the association between two words:

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log(P(w_i, w_j) + \epsilon)} \quad (4)$$

NPMI values go from -1 (never co-occurring words) to +1 (always co-occurring), while the value of 0 suggests complete independence. The most common implementation of C_{NPMI} applies NPMI as in Eqn (4) to couples of words, computing their joint probabilities using sliding windows.

This measure can be applied also to word sets. This is made possible using a vector representation in which each feature consists in the NPMI computed between w_i and a word in the corpus W , according to the formula:

$$\vec{v}(w_i) = \{NPMI(w_i, w_j) | w_j \in W\} \quad (5)$$

The vectors related to each word of the topic are then compared using the cosine similarity C_V .

Fang et al. (2016) introduce **Word Embeddings-based Coherence**. This metric relies on pre-trained word embeddings such as GloVe or word2vec and evaluate the topic quality using a similarity metric between its top words. In other words, a high mutual embedding similarity between a model's top words reflects its underlying semantic coherence. In this paper, we will use the sum of mutual cosine similarity computed on the Glove vectors² of the top 10 words of each topic.

$$C_{WE} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \cos(v_i, v_j) \quad (6)$$

where v_i and v_j are the GloVe vectors of the words w_i and w_j .

In practice, these metrics are computed at topic level and then aggregated using the arithmetic mean, in order to provide a coherence value for the whole model.

²We use a Glove model pre-trained on Wikipedia 2014 + Gigaword 5, available at <https://nlp.stanford.edu/projects/glove/>

3.2 Metrics Which Relies on a Ground Truth

The most used metric that relies on a ground truth is the **Purity**, defined as the fraction of documents in each cluster with a correct prediction (Hajjem and Latiri, 2017). A prediction is considered correct if the original label coincides with the original label of the majority of documents falling in the same topic prediction. Given L the set of original labels and T the set of predictions:

$$Purity(T, L) = \frac{1}{|T|} \sum_{i \in T} \max_{j \in L} |T_j \cap L_j| \quad (7)$$

Other metrics are used in the literature for evaluating the quality of classification or clustering algorithms, applied to the topic modelling task:

1. **Homogeneity**: a topic model output is considered homogeneous if all documents assigned to each topic belong to the same ground-truth label (Rosenberg and Hirschberg, 2007);
2. **Completeness**: a topic model output is considered complete if all documents from one ground-truth label fall into the same topic (Rosenberg and Hirschberg, 2007);
3. **V-Measure**: the harmonic mean of Homogeneity and Completeness. A V-Measure of 1.0 corresponds to a perfect alignment between topic model outputs and ground truth labels (Rosenberg and Hirschberg, 2007);
4. **Normalized Mutual Information (NMI)** is the ratio between the mutual information between two distributions – in our case, the prediction set and the ground truth – normalised through an aggregation of those distributions' entropies (Lancichinetti et al., 2009). The aggregation can be realised by selecting the minimum/maximum or applying the geometric/arithmetic mean. In the case of arithmetic mean, NMI is equivalent to the V-Measure.

In this work, we use their implementations as provided by scikit-learn (Pedregosa et al., 2011).

4 Datasets

A common pre-processing is performed on the datasets before training, consisting of:

- Removing numbers, which, in general, do not contribute to the broad semantics of the document;

- Removing the punctuation and lower-casing the text;
- Removing the standard English stop words;
- Lemmatisation using Wordnet, in order to deal with inflected forms as they are a single semantic item;
- Ignoring words with 2 letters or less. In facts, they are mainly residuals from removing punctuation – e.g. stripping punctuation from *people's* produces *people* and *s*.

The same pre-processing is also applied to the text before topic prediction.

4.1 20 NewsGroups

The 20 NewsGroups collection (20NG) (Lang, 1995) is a popular dataset used for text classification and clustering. It is composed of English news documents, distributed fairly equally across 20 different categories according to the subject of the text. We use a reduced version of this dataset³, which excludes all the documents composed by the sole header while preserving an even partition over the 20 categories. This reduced dataset contains 11,314 documents. We pre-process the dataset in order to remove irrelevant metadata – consisting of email addresses and news feed identifiers – keeping just the textual content. The average number of words per document is 142.

4.2 Agence France Presse

The Agence France Presse (AFP) publishes daily up to 2000 news articles in 5 different languages⁴, together with some metadata represented in the NewsML XML-based format. Each document is categorised using one or more subject codes, taken from the IPTC NewsCode Concept vocabulary⁵. In case of multiple subjects, they are ordered by relevance. In this work, we only consider the first level of the hierarchy of the IPTC subject codes.

From this huge amount of publications, we extracted a dataset containing 125,516 news documents in English released in 2019, with 237 words per document on average.

4.3 Yahoo! Answers Comprehensive Q&A

The Yahoo! Answers Comprehensive Q&A (later simply *Yahoo*) contains over 4 million questions and their answers, as extracted from the Yahoo!

³<https://github.com/selva86/datasets/>

⁴<http://medialab.afp.com/afp4w/>

⁵<http://cv.iptc.org/newscodes/subjectcode/>

Answers website⁶. Each question comes with meta-data such as title, date, and category, as well as a list of user-submitted answers. We construct documents by concatenating the title, body and best answer for each question – following Zhang et al. (2015) – and preprocess the documents in the same way as mentioned above. Then we create 2 subsets:

- *Yahoo balanced*, in which each category is represented by the same number of documents (1000) for a total of 26,000 documents;
- *Yahoo unbalanced*, in which the number of documents sampled from each category is proportional to its presence in the overall dataset, for a total of 22,121 documents.

These two subsets have been realised having a number of document with the same order of magnitude, in order to compare the differences in performance with a balanced and unbalanced sets.

Table 1 summarises the properties of these datasets. The datasets present multiple differences, namely the size, the length of the documents and the distribution of documents per topic (i.e. ground truth label).

5 Experiment and Results

Evaluating an unsupervised task such as Topic Modelling is inherently challenging, and despite the variety of metrics, it is still an open problem (Chang et al., 2009; Blei, 2012). While intrinsic metrics (coherence) try to measure the underlying quality of the topical clusters generated by each model, they do not always match with human judgement. Two very coherent topics (according to the metric) can still fall under the same topic label for a human, and vice-versa, topic models aim to maximise the posterior probability of a document belonging to a coherent topic, regardless of how it maps to human-perceived categories. For instance, *Christianity* and *Atheism* can be both filed as two independent topics or one topic (*religion*) by a human annotator, and while neither arbitrary option is wrong, it constitutes a big difference to how we would evaluate the topic modelling algorithms. They have no means of inferring what humans find to be *topically distinct* beyond co-occurrence statistics, making the comparison to human-annotated labels (as a “gold standard”) quite insufficient. Because of these challenges, few works in the literature (O’Callaghan et al., 2015; Alexander and Gleicher, 2016; Alghamdi and Alfalqi, 2015; Qiang

⁶<https://answers.yahoo.com>

Dataset	# Documents	# Labels	# Documents/label (std)	Document Length (std)
20 NEWSGROUPS	11314	20	565 (56)	122 (241)
AFP	125516	17	4932 (8920)	242 (234)
YAHOO! ANSWERS (BALANCED)	26000	26	1000 (0)	43 (47)
YAHOO! ANSWERS (UNBALANCED)	22121	26	850 (726)	43 (46)

Table 1: Characteristics of the studied datasets

et al., 2020) go beyond simple comparisons that only use one metric or dataset, eclipsing the merits and shortcomings of the other methods. We attempt to provide a more thorough comparison using multiple evaluation datasets (varying in size, document length, number of topics, and label distribution) and metrics from the literature as a step towards a better understanding of the available options and their usability for different potential use-cases.

5.1 Varying the datasets

This section reports a comparison between 9 topic modelling algorithms described in Section 2. Our experimental setup goes as follows:

- For each dataset, we pre-process every document using the process described in Section 4;
- We train each topic model on each dataset, selecting the hyper-parameters through an optimisation process based on grid search, in order to maximise the C_{NPMI} score. The use of a coherence metric as an optimisation objective is justified by the common use-case scenario, in which ground-truth labels are not present. The full set of parameters is documented in the repository⁷;
- For each trained model, we compute all the intrinsic (coherence) metrics and the ground-truth-based ones.

For the experiment, we rely on a topic modelling API⁸. This framework provides a common interface for training, performing topic inference, and evaluating using coherence and ground truth. It includes all the metrics described above.

The number of topics – which must be provided in input to the algorithm for training – has been set to 20, 17 and 26 respectively when training on 20NG, AFP, and Yahoo, in order to mimic the original number of labels in each corpus, except for HDP, which automatically infers the number of topics. For the first two datasets, we perform another training using the same hyper-parameters

⁷Anonymized for double blind purpose: <https://bit.ly/2GM2jFO>

⁸Temporary repository to respect the double-blind period: <http://193.55.113.124/topic-model-api/>

but increasing the number of topics to 50, to study its effect on the performance on the various metrics.

While all the obtained results are available in the repository⁹, we will report in Figure 1 a selection of the most noticeable scores, namely C_{NPMI} , Word Embeddings coherence and V-Measure.

C_{NPMI} values are in line with all the other coherence metrics in terms of ranking (listed in the appendix for brevity), i.e. LDA shows consistently good coherence scores across all datasets, followed by NMF and PVTM.

For the CTM model, we obtained a significantly lower coherence value than the one reported by Bianchi et al. (2020). Further investigation and experiments revealed the impact in this of an additional preprocessing step which reduces the vocabulary to the 2000 most-frequent words. This further preprocessing improves the NPMI score of CTM from -0.028 to 0.116 , while lowering the one of LDA from 0.133 to 0.126 . This confirms the limits of topic modelling comparison and enforces the call for a standard procedure.

Word embeddings coherence demonstrated a better correlation which human judgement (Fang et al., 2016). Unsurprisingly, the two models that rely on word embeddings (LFTM, PVTM) tend to perform notably better (Figure 1).

The V-measure results included in Figure 1 are particularly relevant for understanding the correlation between the predicted topics and the ground truth, as it summarises three metrics – homogeneity, completeness and purity. This metric relies on human choices (either by the editors for AFP or the website users for 20NG and Yahoo) and so it approximates the correlation between the topics as decided by the algorithms and the human (subjective) judgement on the same matter. Again, LDA is leading in overall performances, while other models – LFTM, PVTM, GSDMM – have good scores on particular datasets. The Yahoo dataset is particularly challenging for all models (the maximum value for V-measure is 0.33 for LDA), as compared

⁹<http://193.55.113.124/topic-model-api/appendix.pdf>

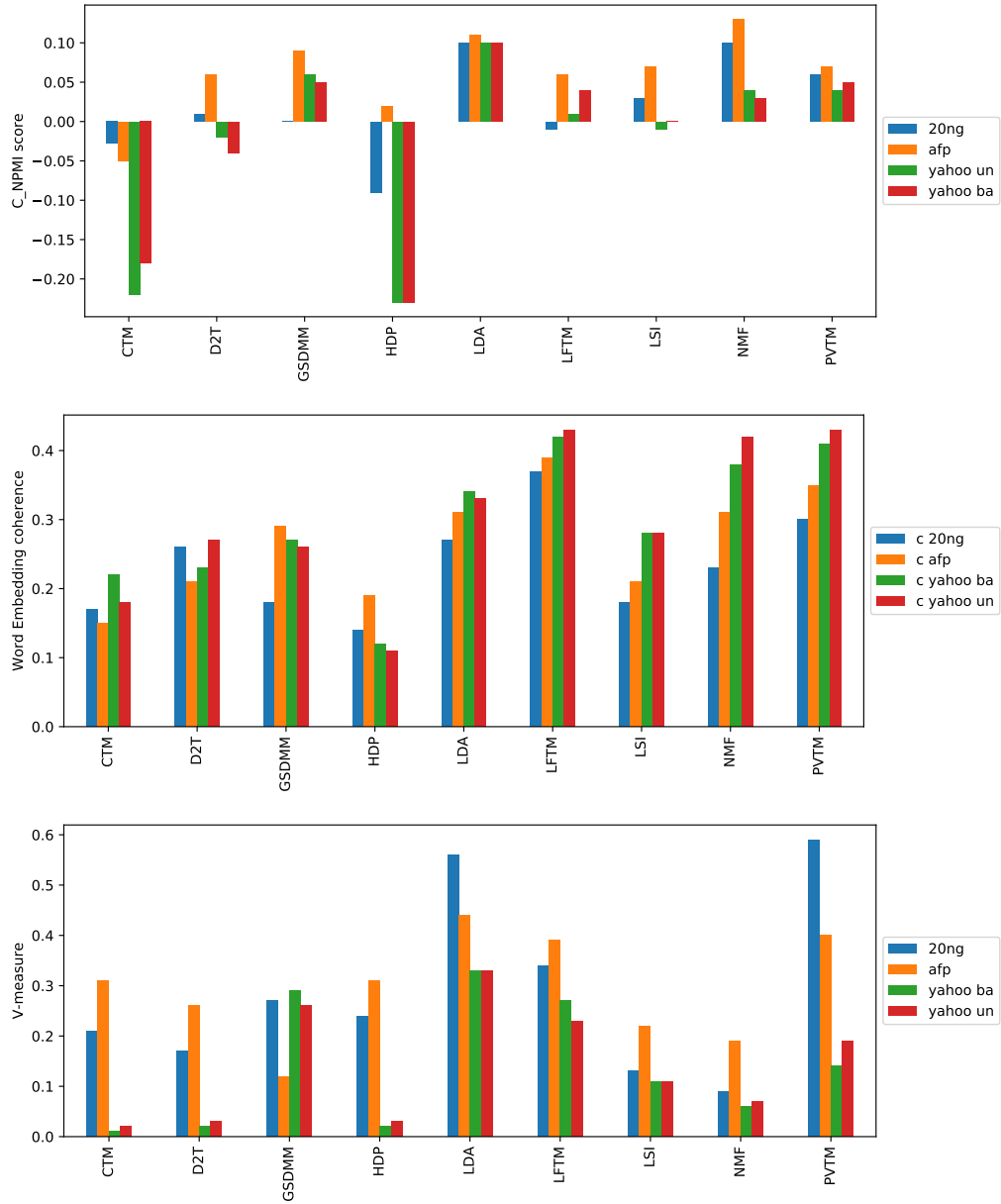


Figure 1: NPMI, Word embedding coherence and V-measure across the models trained on the different datasets.

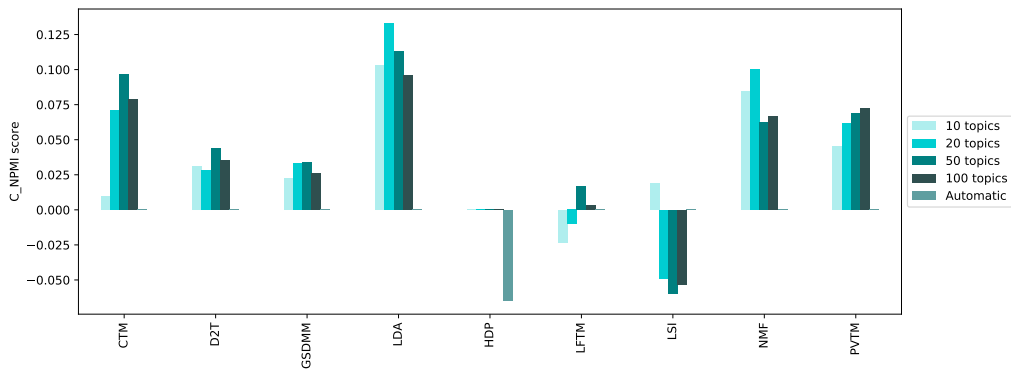


Figure 2: NPMI of each model on the 20NG dataset when varying the number of topics.

to AFP (0.55 for LDA) or 20NG (0.59 for PVTM), probably due to a combination of document length, noise and errors in user-submitted content, and the potential overlap in topics¹⁰. Increasing the number of topic systematically improves the results on AFP, raising the Homogeneity and Purity scores. This happens because the more granular a topic is, the more chance its mapping to the human label is correct. However, this is not observed on 20NG. Given the difference in size between 20NG and AFP, we conclude that the dimension of the former is not allowing it to extract smaller coherent topics, but rather causes an over-specialisation of them.

In summary, LDA still achieves the best scores overall, being often the first (or among the firsts) in ranking for every metric, whereas the other algorithms excel in particular contexts and can be specifically suitable for a given dataset. Increasing the number of topics is particularly helpful on bigger datasets, as it allows the topic models to find smaller yet more coherent subtopics within the collection, avoiding the drawback effect of being too specific. About label balance as tested through the Yahoo dataset, it appears that the balancing in the dataset have not large impact in final results. On the contrary, training on the unbalanced version is often producing better coherence and V-measure. The reason of this can be found in the complete dropping of smaller categories, thus reducing the number of classes and achieving a higher-scoring topic/label mapping.

5.2 Varying the number of topics

To evaluate the effect of the choice of the number of topics (usually unknown beforehand), we train our models – except HDP, which infers the number of topics automatically – on 20NG using the same hyperparameters and varying only the number of topics. The results are shown in Table 2.

While there is a slight yet consistent improvement in the NPMI score for PVTM, we observe that increasing the number of topics does not consistently improve or hurt the coherence of the produced models. The fact that the score for 20 topics is usually the highest is probably due to the model finetuning, applied on this configuration. Finetuning every model for every number of topics requires a study of the co-optimisation of hyperparameters, which is out of the scope of this paper.

¹⁰Some examples are “News & Events”/“Politics and Government”, “Dining Out”/“Food & Drink”, and “Business and Finance”/“Local Businesses”

NPMI	Mean (std)	Max	Min
HDP	-0.176 (0.09)	-0.06	-0.28
LDA	0.120 (0.01)	0.133	0.101
NMF	0.083 (0.01)	0.102	0.063
PVTM	0.054 (0.01)	0.061	0.046

Table 2: The effect of random seeds on the NPMI for some models trained on 20NG

5.3 Varying the seed

For the models which we able to configure the random seed, we perform the evaluation on 20NG using the same hyperparameters except the seed (which we varied to have the values from 1 to 5). Even among 5 runs, we observe quite some variance in the metrics that is purely due to randomness which can be quite substantial. We report these results in Figure 2.

While the effect is not very pronounced, it can be misleading. We thus recommend for topic models relying on random initialization to evaluate their models using different seeds, to guarantee a statistically significant comparison.

6 Conclusions and Future Work

In this work, we empirically compared 9 topic modelling algorithms using different coherence and ground-truth-based metrics on 3 text corpora reflecting a variety of properties, using a common evaluation framework. The results reveal several differences between the trained models, which obtain more or less better performances in specific settings. Among these, LDA proves to be the most consistent resulting coherence, while embedding-based models prove to be less prone to generate meaningless topics.

The task of evaluating topic modelling remains a challenging one because of the inherent lack of a ground-truth, the subjectivity of what constitutes a topic, and the variety of settings wherein it is used. While every newly proposed topic model claims to improve on the existing state-of-the-art under some specific conditions, it is a worthwhile effort to revisit those claims and review them on a broader set of challenges and a unified pipeline, revealing their strengths and shortcomings.

This study focusing on quantitative measures, a possible extension may involve involving human evaluation with domain experts, in order to judge the quality of the predicted topics as well as their relevance. Future work may also investigate the

impact of other variables such as language, the length of documents and the size of the dataset on the overall performance of each studied model.

References

- Eric Alexander and Michael Gleicher. 2016. [Task-Driven Comparison of Topic Models](#). *IEEE Transactions on Visualization and Computer Graphics*, 22(1):320–329.
- Rubayyi Alghamdi and Khalid Alfalqi. 2015. [A survey of topic modeling in text mining](#). *International Journal of Advanced Computer Science and Applications*, 6.
- Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2020. Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. *arXiv preprint arXiv:2004.03974*.
- David M. Blei. 2012. [Probabilistic topic models](#). *Commun. ACM*, 55(4):77–84.
- David M. Blei and Jon D. McAuliffe. 2007. Supervised Topic Models. In *20th International Conference on Neural Information Processing Systems (NIPS)*, page 121–128, Red Hook, NY, USA. Curran Associates Inc.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.
- Sophie Burkhardt and Stefan Kramer. 2019. [A survey of multi-label topic models](#). *SIGKDD Explor. Newsl.*, 21(2):61–79.
- Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A Novel Neural Topic Model and Its Supervised Extension. In *AAAI Conference on Artificial Intelligence*.
- Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. [Reading tea leaves: How humans interpret topic models](#). In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc.
- Christian Chiarcos, Richard Eckart de Castilho, and Manfred Stede. 2009. *Von der Form zur Bedeutung: Texte automatisch verarbeiten - From Form to Meaning: Processing Texts Automatically*. Narr Francke Attempto Verlag GmbH + Co. KG.
- Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.
- Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. 2016. [Using Word Embedding to Evaluate the Coherence of Topics from Twitter Data](#). In *39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16*, page 1057–1060, New York, NY, USA. Association for Computing Machinery.
- Thomas S. Ferguson. 1973. [A bayesian analysis of some nonparametric problems](#). *Ann. Statist.*, 1(2):209–230.
- Derek Greene, Derek O’Callaghan, and Pádraig Cunningham. 2014. How Many Topics? Stability Analysis for Topic Models. In *Machine Learning and Knowledge Discovery in Databases*, pages 498–513, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Malek Hajjem and Chiraz Latiri. 2017. Combining IR and LDA Topic Modeling for Filtering Microblogs. In *21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES)*, volume 112, pages 761 – 770, Marseille, France.
- Hamed Jelodar, Yongli Wang, Chi Yuan, and Xia Feng. 2017. [Latent dirichlet allocation \(LDA\) and topic modeling: models, applications, a survey](#). *CoRR*, abs/1711.04305.
- Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. [Detecting the overlapping and hierarchical community structure in complex networks](#). *New Journal of Physics*, 11(3):033015.
- Ken Lang. 1995. NewsWeeder: Learning to Filter News. In *20th International Conference on Machine Learning*, pages 331 – 339, San Francisco, USA. Morgan Kaufmann.
- Quoc Le and Tomas Mikolov. 2014. [Distributed representations of sentences and documents](#). In *31st International Conference on Machine Learning Research*, volume 32, pages 1188–1196, Beijing, China. PMLR.
- David Lenz and Peter Winker. 2020. [Measuring the diffusion of innovations with paragraph vector topic models](#). *PLOS ONE*, 15(1):1–18.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *26th International Conference on Neural Information Processing Systems (NIPS)*, volume 2, pages 3111–3119, Lake Tahoe, NV, USA. Curran Associates Inc.
- David Newman, Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2006. Analyzing Entities and Topics in News Articles Using Statistical Topic Models. In *Intelligence and Security Informatics*, pages 93–104, Berlin, Heidelberg. Springer.

- David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010a. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 100–108, USA. Association for Computational Linguistics.
- David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010b. [Evaluating topic models for digital libraries](#). In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, page 215–224, New York, NY, USA. Association for Computing Machinery.
- Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving Topic Models with Latent Feature Word Representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.
- Derek O’Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. 2015. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645 – 5657.
- Pentti Paatero and Unto Tapper. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. 2020. [Short Text Topic Modeling Techniques, Applications, and Performance: A Survey](#). *IEEE Transactions on Knowledge and Data Engineering*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-](#)
[networks](#). In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *8th ACM International Conference on Web Search and Data Mining (WSDM)*, page 399–408, New York, USA. ACM.
- Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.
- Alexandra Schofield and David Mimno. 2016. [Comparing apples to apple: The effects of stemmers on topic models](#). *Transactions of the Association for Computational Linguistics*, 4:287–300.
- Akash Srivastava and Charles Sutton. 2017. Autoencoding Variational Inference For Topic Models. In *ICLR*.
- Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. [Hierarchical dirichlet processes](#). *Journal of the American Statistical Association*, 101(476):1566–1581.
- Xing Yi and James Allan. 2009. A Comparative Study of Utilizing Topic Models for Information Retrieval. In *Advances in Information Retrieval*, pages 29–41, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Jianhua Yin and Jianyong Wang. 2014. A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering. In *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, page 233–242, New York, USA. ACM.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

B.5 EURECOM's ESWC 2021 poster paper

This paper describes the GraphNER idea proposed by EURECOM and accepted at ESWC 2021.

Named Entity Recognition as Graph Classification

Ismail Harrando and Raphaël Troncy

EURECOM, Sophia Antipolis, France
{ismail.harrando,raphael.troncy}@eurecom.fr

Abstract. Injecting real-world information (typically contained in Knowledge Graphs) and human expertise into an end-to-end training pipeline for Natural Language Processing models is an open challenge. In this preliminary work, we propose to approach the task of Named Entity Recognition, which is traditionally viewed as a *Sequence Tagging* problem, as a *Graph Classification* problem, where every word is represented as a node in a graph. This allows to embed contextual information as well as other external knowledge relevant to each token, such as gazetteer mentions, morphological form, and linguistic tags. We experiment with a variety of graph modeling techniques to represent words, their contexts, and external knowledge, and we evaluate our approach on the standard CoNLL-2003 dataset. We obtained promising results when integrating external knowledge through the use of graph representation in comparison to the dominant end-to-end training paradigm.

Keywords: Named Entity Recognition · Knowledge Graph · Graph Classification · Knowledge Injection.

1 Introduction

Transformer-based language models such as BERT [2] have tremendously improved the state of the art on a variety of Natural Language Processing tasks and beyond. While it is hard to argue against the performance of these language models, taking them for granted as the fundamental building-block for any NLP application stifles the horizon of finding new and interesting methods and approaches to tackle quite an otherwise diverse set of unique challenges related to specific tasks. This is especially relevant for tasks that are known to be dependent on real-world knowledge or domain-specific and task-specific expertise. Although these pre-trained language models have been shown to internally encode some real-world knowledge (by virtue of being trained on large and encyclopedic corpora such as Wikipedia), it is less clear which information is actually learnt and how it is internalized, or how one can inject new external information (e.g. from a knowledge base) into these models in a way that it does not require retraining them from scratch.

In this work, we propose a novel method to tackle Named Entity Recognition, a task that has the particularity of relying on both the linguistic understanding

of the sentence as well as some form of real-world information, as what makes a Named Entity is the fact that it refers to an entity that is generally designated by a proper name. Since graphs are one of the most generic structures to formally represent knowledge (e.g. Knowledge Graphs), they constitute a promising representation to model both the linguistic (arbitrarily long) context of a word as well as any external knowledge that is deemed relevant for the task to perform. Graph connections between words and their descriptions seems to intuitively resemble how humans interpret words in a sentence context (how they relate to preceding and following words, and how they relate external memorized knowledge such as being a "city name" or "an adjective"). Hence, we propose to cast Named Entity Recognition as a Graph Classification task, where the input of our model is the representation of a graph that contains the word to classify, its context, and other external knowledge modeled either as nodes themselves or as node features. The output of the classification is a label corresponding to the entity type of the word (Figure 1).

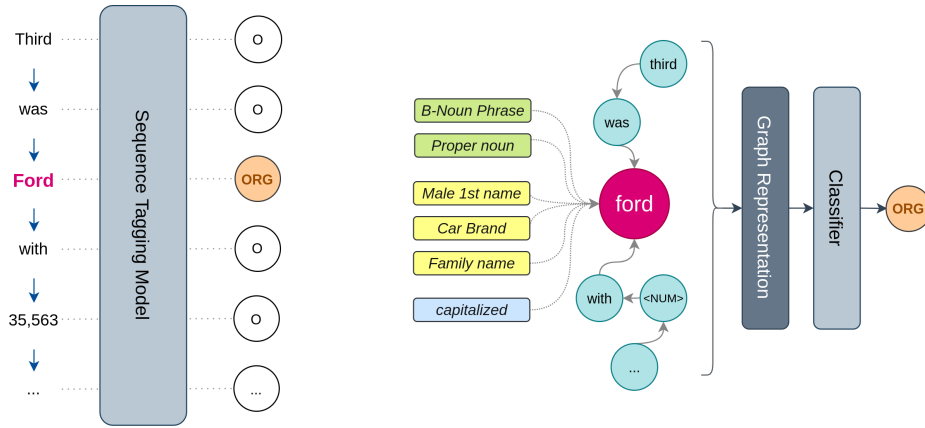


Fig. 1. Left: Traditional sequence tagging model. Right: Each *word* in a sentence becomes the central node of a graph, linked to the words from its context, as well as other task-related features such as grammatical properties (e.g. "Proper Noun"), gazetteers mentions (e.g. "Car Brand") and task-specific features (e.g. "Capitalized"). The graph is then embedded which is passed to a classifier to predict an entity type.

2 Approach

In order to perform Named Entity Recognition as a Graph Classification task, the "word graph" needs to be transformed into a fixed-length vector representation, that is then fed to a classifier (Figure 1). This graph representation needs to embed the word to classify (the *central node*), as well as its *context* – words appearing before and after it – and its related *tags* (properties such as gazetteers mention, grammatical role, etc). This formalization is interesting because it allows to represent the entire context of the word (as graphs can be arbitrarily

big), to explicitly model the left and the right context separately, and to embed different descriptors to each word seamlessly (either as node features or as other nodes in the graph) and thus help the model to leverage knowledge from outside the sentence and the closed training process. This is a first difference with the traditional sequence labeling methods that only consider a narrow window the tokens to annotate. While we posit that this method can integrate any external data in the form of new nodes or node features in the input graph, we focus on the following properties that are known to be related to the NER task:

- **Context:** which is made of the words around the word we want to classify.
- **Grammatical tags:** we use the Part of Speech (POS) e.g. ‘Noun’, as well as the shallow parsing tags (**chunking**) e.g. ‘Verbal Phrase’.
- **Case:** in English, capitalization is an important marker for entities. We thus add tags such as: ‘Capitalized’ if the word starts with a capital letter, ‘All Caps’ if the word is made of only uppercase letters, and so on.
- **Gazetteers:** we generate lists of words that are related to potential entity types by querying Wikidata for labels and synonyms corresponding to entities belonging to types of interest such as *Family Name*, *Brand*, etc.

The literature on Graph Representations shows a rich diversity in approaches [1, 3], but for our early experiments, we choose one candidate from each of the main representation families: a neural auto-encoder baseline, Node2Vec for node embeddings, TransE for Entity Embeddings, and a Graph Convolutional Network based on [3]. This is admittedly a small sample of the richness that can be further explored in the future, both in terms of the models and the way the input graph is constructed (how to model the context and the added knowledge).

3 Experiments and Results

3.1 Experimental protocol

To train each of the aforementioned models, we construct a dataset¹ by going through every word in every document from the CoNLL training dataset, and build its graph (Figure 1). Each of these graphs is then turned into a fixed-length vector that is fed to a neural classifier (see subsection 3.2). For each of the representations, we fine-tune the hyper-parameters (e.g. the embedding size) using the CoNLL validation (dev) set. We report the Micro-F1 and Macro-F1 scores for all trained models in Table 1 for both the validation and the test sets together with the currently best performing approach from the state of the art².

3.2 Methods

To evaluate the approach, we selected the following methods to generate graph embeddings:

¹ https://github.com/Siliam/graph_ner/tree/main/dataset/conll

² See also http://nlpprogress.com/english/named_entity_recognition.html

1. **Binary Auto-encoder:** We represent the input graph as a binary embedding of the different nodes that are present in it, i.e., we concatenate a one-hot embedding of the word, its left context and right context separately, and one-hot embeddings for all other extra tags in the vocabulary (e.g. POS tags). We use this "flat" representation of the graph as a baseline that incorporates all the external data without relying on the graph connectivity. We first train a neural encoder-decoder (both feed-forward neural networks with one hidden layer) to reconstruct the input binary representation of the graph, then use the encoder part to generate a the graph embedding.
2. **Node2Vec:** we generate a global graph representing all nodes in the training set (all words as related to their context, plus the tags that we also represent as nodes), and then we use **Node2Vec** to generate embeddings for all nodes in our graph. The final input graph representation is obtained by averaging all nodes representations (i.e. the word, its context and its tags).
3. **TransE:** same as for **Node2Vec**, except the edges between the different nodes (entities) are now labeled relations e.g. 'before', 'after', 'pos', etc. We average the representations of each of these nodes to obtain a graph embedding.
4. **GCN:** unlike the previous approaches where a graph embedding is generated before the training phase, we can directly feed the graph data into a GCN and train it end-to-end, thus allowing the network to learn a task-specific graph representation. We base our model on GraphSAGE-GCN [3], using an architecture based on this model from the PyTorch Geometric Library³ that we modify to account for additional node features (tags, gazetteers classes etc). This allows the network to learn a graph representation that is specific to this task.

We note that for all of these methods, the classifier is a fully-connected neural network with 1 hidden layer, and we add weights to the loss function to accommodate for the unbalance in label distribution based on this formula:

$$w_{label_i} = \sqrt{\frac{\min(count(label_j) \text{ for } label_j \text{ in labels})}{count(label_i)}}$$

3.3 results

We observe a significant decrease in performance for all models between the evaluation and test sets (with a varying intensity depending on the choice of the model) that is probably due to the fact that the test set contains a lot of out-of-vocabulary words that do not appear in the training set. We also see that adding the external knowledge consistently improve the performance of the graph models on both Micro- and Macro-F1 for all models considered. Finally, while the performance on the test set for all graph-only models is still behind LUKE, the best performing state of the art NER model on ConLL 2003, we

³ https://github.com/rusty1s/pytorch_geometric/blob/master/examples/proteins_topk_pool.py

observe that these models are significantly smaller and thus faster to train (in matters of minutes once the graph embeddings are generated), when using a simple 2-layers feed-forward neural as a classifier. These preliminary results show promising directions for additional investigations and improvements.

Method	Dev m-F1	Dev M-F1	Test m-F1	Test M-F1
Auto-encoder	91.0	67.3	90.3	63.2
Auto-encoder+	91.5	71.7	91.5	70.4
Node2Vec	93.3	81.6	90.0	68.3
Node2Vec+	94.1	82.1	91.1	72.6
TransE	91.8	75.0	91.7	70.0
TransE+	93.6	78.8	91.9	74.5
GCN	96.1	86.3	92.9	78.8
GCN+	96.5	88.8	94.1	81.0
LUKE [4]			94.3	

Table 1. NER results with different graph representations (CoNLL-2003 dev and test sets). The entries marked with “+” represent the models with external knowledge added to the words and their context.

4 Conclusion and Future work

While the method proposed in this paper shows some promising results, the performance on the ConLL 2003 test set is still significantly lower than the best state-of-the-art Transformer-based method as of today. However, we have made multiple design choices to limit the models search space and we believe that additional work on the models themselves (different architectures, hyper-parameters fine-tuning, adding attention, changing the classifier) can improve the results. The drop of performance from the validation to the test set is probably due to the lack of any external linguistic knowledge outside of the training set, which can be overcome by enriching the nodes with linguistic features such as Word Embeddings. We will test this method on other specialized datasets in order to demonstrate the value of this approach for domain-specific applications (fine-grained entity typing). To facilitate reproducibility, we published the code of our experiments at https://github.com/Siliam/graph_ner.

References

1. Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., Murphy, K.: Machine Learning on Graphs: A Model and Comprehensive Taxonomy. arxiv 2005.03675 (2021)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT (2019)
3. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive Representation Learning on Large Graphs. In: NIPS (2017)
4. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In: EMNLP (2020)

B.6 EURECOM's DataTV 2021a workshop paper

This paper describes the FaceRec open source library developed by EURECOM and accepted at DataTV 2021 colocated with IMX 2021.

FaceRec: An Interactive Framework for Face Recognition in Video Archives

PASQUALE LISENA, EURECOM, France

JORMA LAAKSONEN, Aalto University, Finland

RAPHAËL TRONCY, EURECOM, France

Annotating the visual presence of a known person in a video is a hard and costly task, in particular when applied to large video corpora. The web is a massive source of visual information that can be exploited for detecting celebrities. In this work, we introduce FaceRec, an AI-based system for automatically detecting faces of known but also unknown people in a video. The system relies on a combination of state-of-the-art algorithms (MTCNN and FaceNet), applied on images crawled from web search engines. A tracking system links consecutive detection in order to adjust and correct the label predictions using a confidence-based voting mechanism. Furthermore, we add a clustering algorithm for the unlabelled faces, thus increasing the number of people that can be recognized. We evaluate our system that obtained high precision on datasets of both historical and recent videos. We release the complete framework as open-source at <https://git.io/facerec>.

CCS Concepts: • **Computing methodologies** → **Object recognition**; • **Information systems** → *Web mining*; • **Computer systems organization** → *Neural networks*.

Additional Key Words and Phrases: Face recognition, neural networks, semantic metadata, video archives

ACM Reference Format:

Pasquale Lisena, Jorma Laaksonen, and Raphaël Troncy. 2021. FaceRec: An Interactive Framework for Face Recognition in Video Archives. In *Proceedings of 2nd International Workshop on Data-driven Personalisation of Television (DataTV-2021)*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>



Fig. 1. Charles de Gaulle and Dwight D. Eisenhower together in 1962 (picture from Archives Nationales).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

Manuscript submitted to ACM

1 INTRODUCTION

Identifying people appearing in videos is undoubtedly an important cue for automatically understanding video content. Knowing who appears in a video, when and where, can also lead to learning interesting patterns of relationships among characters, with interesting applications in historical research and media discovery. Such person-related annotations enable to generate more accurate segmentation and more compelling video descriptions, facilitating multimedia search and re-use of video content. Media archives contain numerous examples of celebrities appearing in the same news segment (Figure 1). However, the annotations produced manually by archivists do not always identify with precision those individuals in the videos. The presence of digital annotations is particularly crucial for large corpora, whose metadata are the only efficient way to identify relevant elements [22]. At the same time, relying on human annotations is not a scalable solution when handling large volumes of video resources.

The web offers an important amount of pictures of people and in particular of celebrities, easily findable using their full name as search terms in a general purpose search engine such as Google. While it has been considered a relevant information source in other communities – such as computational linguistics [13] and recommender system [17] – the web is still only scarcely exploited in image analysis and in face recognition in particular.

In this work, we aim to leverage pictures of celebrities crawled from the web for identifying faces of people in video archives. In doing so, we develop FaceRec, an interactive framework for face recognition in video corpora that relies on state-of-the-art algorithms. The system is based on a combination of MTCNN (face detection) and FaceNet (face embeddings), whose vector representations of faces are used to feed a classifier, which is then used to recognise faces at the frame level. A tracking system is included in order to increase the robustness of the library towards recognition errors in individual frames for getting more consistent person identifications.

The rest of this paper is organised as follows. After reporting some relevant work in Section 2, we describe our approach in Section 3. A quantitative evaluation is carried out on two different datasets in Section 4. We introduce the FaceRec API and a web application for visualizing the results in Section 5. Finally, some results and possible future work are outlined in Section 6.

2 RELATED WORK

During the last decade, there has been substantial progress in the methods for automatic recognition of individuals. The recognition process generally consists of two steps. First, faces need to be detected in a video, i.e. which region of the frame may contain a face. Second, those faces should be recognised, i.e. to whom a face belongs.

The Viola-Jones algorithm [21] for face detection and the Local Binary Pattern (LBP) features [1] for the clustering and recognition of faces were the most famous methods until the advent of deep learning and convolutional neural networks (CNN). Nowadays, two main approaches are used for detecting faces in video and both use CNNs. One implementation is available in the Dlib library [14] and provides good performance for frontal images, but it requires an additional alignment step before the face recognition step can be performed. The recent Multi-task Cascaded Convolutional Networks (MTCNN) [24] approach provides even better performance using an image pyramid approach and using face landmarks detection for re-aligning the detected faces to the frontal orientation.

After locating the position and orientation of the faces in the video frames, the face recognition process can be performed. There are several strategies available in the literature for face recognition. Currently, the most practical approach is to perform face comparison using a transformation space in which similar faces are mapped close together,

and to use this representation to identify individuals. Such embeddings, computed on large collections of faces have been made available to the research community, such as the popular FaceNet [19].

In [23], MTCNN and FaceNet are used in combination and tested with eight public face datasets, reaching a recognition accuracy close to 100% and surpassing other methods. These results have been confirmed in several surveys [8, 20] and in recent works [2]. In addition, MTCNN has been recognised to be very fast while having good performance [16].

Given the almost perfect performance of the MTCNN + FaceNet face recognition setups, our work focuses on setting up a complete system built upon these technologies. In this perspective, our contribution does not consist of a new state-of-the-art performance in face recognition, but of the combination and application of available techniques in combination with images crawled on the web.

3 METHOD

This section describes the FaceRec pipeline, detailing the training and the recognition tasks, including the additional strategy for recognising unknown faces in videos.

3.1 Training the system

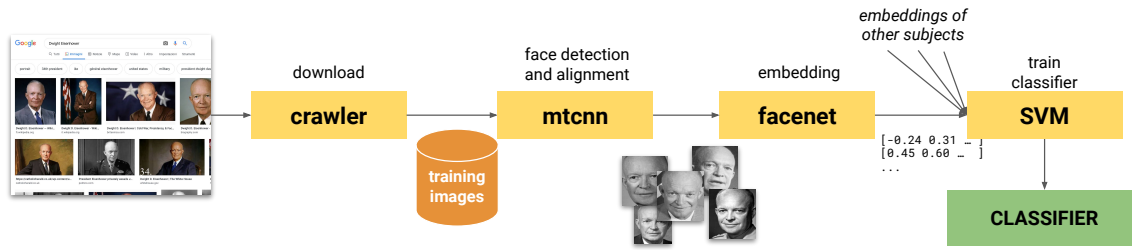


Fig. 2. FaceRec training pipeline

During training, our system retrieves images from the web for realising a face classifier (Figure 2). The first module is a **crawler**¹ which, given a person’s name, automatically downloads a set of k photos using Google’s image search engine. In our experiments, we have typically used $k = 50$. After converting them to greyscale, we apply to each image the **MTCNN algorithm** [24] for face detection². MTCNN returns in output the bounding box of the face in the frame and the position of relevant landmarks, namely the position of eyes, nose and mouth limits. The recognised faces are cropped, resized and aligned in order to have in output a set of face images of width $w = 256$ and height $h = 256$, in which the eyes are horizontally aligned and centered. In particular, the alignment consists of a rotation of the image. Chosen the desired positions for the left (x_l, y_l) and right eye (x_r, y_r) ³ and given their original positions (a_l, b_l) and (a_r, b_r) , the image is rotated by an angle α on the centre c with scale factor s , computed in the following way:

$$dX = a_r - a_l$$

$$dY = b_r - b_l$$

¹We use the *icrawler* open-source library: <https://github.com/hellok/icrawler/>

²We use the implementation at <https://github.com/ipazc/mtcnn>

³We use $x_l = 0.35w$, $x_r = (1 - x_l)$, and $y_l = y_r = 0.35h$.

$$\alpha = \arctan \frac{dY}{dX} - 180^\circ$$

$$c = \left(\frac{x_l + x_r}{2}, \frac{y_l + y_r}{2} \right)$$

$$s = \frac{(x_r - x_l) \cdot w}{\sqrt{dX^2 + dY^2}}$$

Not all resulting cropped images are suitable for training a classifier. They may contain faces of other individuals, if they have been extracted from a group picture or if the original picture was not really depicting the searched person. Other cases which may have a negative impact on the system are side faces, low resolution images, drawings and sculptures. In order to exclude those images, we relied on two complementary approaches, which we used in combination:

- using face embeddings to automatically remove the outliers. This is realised by removing the face with the highest cosine distance from the average embedding vector, until the standard deviation of all differences is under an empirically chosen threshold (0.1);
- allowing the user to further improve the automatic selection by allowing the exclusion of faces via the user interface (Section 5).

On the remaining pictures, a pretrained **FaceNet** [19] model with Inception ResNet v1 architecture trained on the VGGFace2 dataset [6] is applied for extracting visual features or embeddings of the faces. The embedding vectors feed ***n* parallel binary SVM**⁴ classifiers, where *n* is the number of distinct individuals to recognise. Each classifier is trained in a one-against-all approach [12], in which the facial images of the selected individual are used as positive samples, while all the others are considered negative samples. In this way, each classifier provides in output a confidence value, which is independent of the outputs of all other classifiers. This will allow to set – in the recognition phase – a confidence threshold for the candidate identities which does not depend on *n*, making the system scalable⁵.

3.2 Recognising faces in video

The face recognition pipeline is composed of:

- operations that are performed at the frame level and are shown in Figure 3. In order to speed up the computation, it is possible to set a sampling period *T*. For our experiments, we set *T* = 25, in order to process one frame per second;
- operations of synthesis on the results, which take into account the tracking information across frames for providing more solid results.

In each frame, **MTCNN** detects the presence of faces, to which is applied the same cropping and alignment presented in Section 3.1. Their **FaceNet** embeddings are computed and the **classifier** selects the best match among the known faces, assigning a confidence score in the interval [0, 1].

⁴SVM obtained better performance than other tested classifier, namely Random Forest, Logistic Regression and the k-Nearest Neighbours.

⁵We also performed experiments on this system using a multi-class classifier with *n* class, instead of the *n* binary classified. While the results revealed similar precision scores, the recall for the multi-class solution was considerably worse, 22 percentage points lower than the system with binary classifiers.

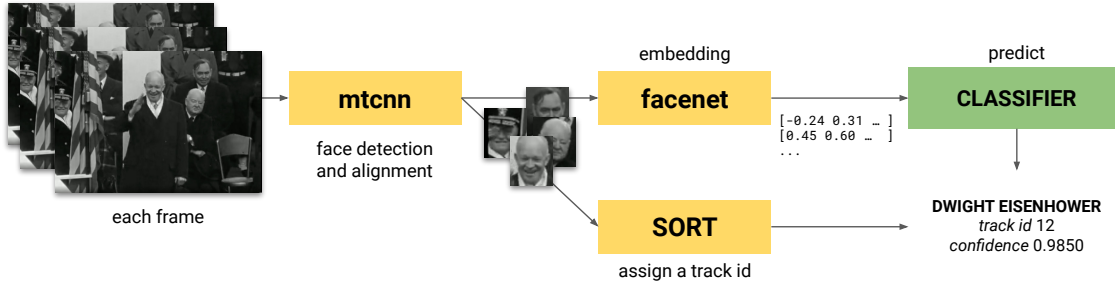


Fig. 3. FaceRec prediction pipeline

At the same time, the detected faces are processed by *Simple Online and Realtime Tracking (SORT)*, an object tracking algorithm which can track multiple objects (or faces) in realtime⁶ [5]. The algorithm uses the MTCNN bounding box detection and tracks the bounding boxes across frames, assigning a tracking id to each face.

After having processed the entire video, we obtain a set of detected faces, each of them with a predicted label, confidence score and tracking id, as well as space and time coordinates. This information is then processed at the level of single tracking collection, integrating the data of the different recognitions having the same tracking id. For a given tracking – including a certain number of samples – we compute the mode⁷ of all predictions, as well as the weighted mode with respect to the confidence scores. A unique predicted label p is chosen including among all the possible predictions if it satisfies all the following conditions:

- p is both the mode and the weighted mode;
- the ratio of samples with prediction p over the total samples is greater than the threshold h ;
- the ratio of samples with prediction p over the total samples, weighting all occurrence with the confidence score, is greater than the threshold h_w .

We empirically found $h = 0.6$ and $h_w = 0.4$ as the best values for the thresholds. It is possible that a tracking does not produce a label fulfilling all the conditions. In that case, the prediction is considered uncertain and the tracking is excluded from the results. We assign to the tracking a unique confidence score from the arithmetic mean of the scores of the sample with prediction p . We intentionally exclude the minority of wrong predictions in this computation: in this way, wrong predictions – caused by e.g. temporary occlusion or turn of the head by side – do not penalise the overall scores. The final results are then filtered again by overall confidence using a threshold t , whose impact is discussed in Section 4.

3.3 Building models for unknown faces

So far, the described system is trained for recognising the faces of known people. During the processing of a video, several detected faces may not be matched with any of the individuals in the training set. However, these people may still be relevant to be tracked and inserted in the list of people to search. Therefore, in addition to the pipeline based on images crawled from the web, a face clustering algorithm is active in the background with the objective of detecting non-celebrities or more simply, any persons not present in the training set. At runtime, all FaceNet features extracted from faces in the video frames are collected. Once the video has been fully processed, these features are aggregated

⁶We used the implementation provided at <https://github.com/Linzaer/Face-Track-Detect-Extract> with some minor modification

⁷The mode is "the number or value that appears most often in a particular set" (*Cambridge Dictionary*)

through hierarchical clustering⁸ based on a distance threshold, empirically set to 14. The clustering produces a variable number m of clusters, with all items assigned to one of them. The clusters are then filtered in order to exclude:

- those for which we can already assign a label from our training set;
- those having a distance — computed as the average distance of the elements from the centroid — larger than a second, more strict threshold, for which we have used the value 1.3;
- those having instances of side faces in the centre of the cluster. In particular, we observed that in those cases, the resulting cluster produces unreliable results and groups profile views of different people.

With MTCNN, we obtain the position of the following landmarks: left eye (a_l, b_l), right eye (a_r, b_r), left mouth corner (m_l, n_l), right mouth corner (m_r, n_r). We compute the ratio r_{dist} between the distance between mouth and eyes and the distance between the two eyes:

$$\begin{aligned} dX &= a_r - a_l & dG &= m_l - a_l \\ dY &= b_r - b_l & dH &= n_l - b_l \\ dist_{wide} &= \sqrt{dX^2 + dY^2} & dist_{high} &= \sqrt{dG^2 + dH^2} \\ r_{dist} &= \frac{dist_{high}}{dist_{wide}} \end{aligned}$$

This value is inversely proportional to the eyes' distance on the image, increasing when the eyes are closer, e.g. in face rotation to a side. We identified as side faces the cases in which $r_{dist} > 0.6$. Finally, only the 5 faces closest to each centroid are kept, in order to exclude potential outliers.

The system returns in output the remaining clusters, which are temporarily assigned to a label of type *Unknown <i>*, where i is an in-video incremental identifier – e.g. *Unknown 0*, *Unknown 1*, etc. The clusters can be labelled with human effort: in this case, the relevant frames are used as training images and the person is included in the training set. This strategy is particularly useful in cases when the crawler module cannot be used to obtain representative samples of the individuals appearing in the videos.

4 EVALUATION

In this section, we evaluate the FaceRec system measuring the precision and recall on two different ground-truth datasets: one of historical videos and one composed of more recent video footage.

4.1 Creation of a ground truth

In the absence of a large and rigorous ground truth dataset of faces in video, we developed two evaluation datasets of annotated video fragments from two different specialised corpora.

ANTRACT dataset. *Les Actualités françaises*⁹ are a series of news programmes broadcasted in France from 1945 to 1969, currently stored and preserved by the *Institute national de l'audiovisuel (INA)*¹⁰. The videos are in black-and-white, with a resolution of 512×384 pixels. Metadata are collected through INA's *Okapi* platform [4, 7], which exposes a SPARQL endpoint.

⁸We used the implementation available in SciPy: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.fcluster.html>

⁹<https://www.ina.fr/emissions/les-actualites-francaises/>

¹⁰The corpus can be downloaded from <https://dataset.ina.fr/>

A list of 13 historically well-known people has been provided by domain experts. From the metadata, we have obtained the reference to the segments in which these people appear and the subdivision of these segments in shots¹¹. This search produced 15,628 shots belonging to 1,222 segments from 702 media. In order to reduce the number of shots and to check manually the presence of the person in the selected segments, we performed face recognition on the central frame of each shot. The final set has been realised with an iteration of automatic sampling and manual correction, adding also some shots not involving any of the specified people. At the end, it includes 198 video shots (belonging to 129 distinct media), among which 159 segments (80%) featured one or more of the 13 known people and 39 segments (20%) did not include any of the specified people.

MeMAD dataset. This dataset has been developed from a collection of news programmes broadcasted on the French TV channel *France 2* in May 2014. These videos – in colour, 455×256 pixels – are part of the MeMAD video corpus¹², with metadata available from the MeMAD’s Knowledge Graph¹³. We followed the same procedure than above with the following differences. In this case, the list of people to search is composed of the six most present ones in the MeMAD Knowledge Graph’s video segments. Without the information about the subdivision in shots, for each segment of duration d , we performed face recognition on the frames at positions $d/4$, $d/2$ and $3d/4$, keeping only the segments with at least one found face. We also made an automatic sampling and a manual correction as for the ANTRACT dataset. The final set includes 100 video segments, among which 57 segments (57%) featured one of the six known people and 43 segments (43%) did not include any of the specified people.

Table 1 summarises the main differences between the two datasets.

	ANTRACT	MeMAD
type	historical images	TV news
years	1945-1969	2014
resolution	512×384	455×256
colourspace	b/w	colour
shots division	yes	no
list of celebrities to search	13 (chosen by domain experts)	6 (most present in KG)
represented fragment and length	shot 3 seconds in avg.	segment up to 2 minutes
records	216	100
distinct fragments	198	100
distinct media (videos)	129	30
fragments without known faces	39	43

Table 1. Description of the ANTRACT and MeMAD datasets

¹¹In the following, we define *media* as the entire video resource (e.g. an MPEG-4 file), *segment* a temporal fragment of variable length (possibly composed of different shots), and *shot*, a not interrupted recording of the video-camera. See also the definitions of MediaResource, Part and Shot in the EBU Core ontology (<https://www.ebu.ch/metadata/ontologies/ebucore/>)

¹²<https://memad.eu/>

¹³<https://data.memad.eu/>

Person	P	R	F	S
Ahmed Ben Bella	1.00	0.46	0.63	13
François Mitterrand	1.00	0.92	0.96	13
Pierre Mendès France	1.00	0.61	0.76	13
Guy Mollet	0.92	0.92	0.92	13
Georges Bidault	0.83	0.71	0.76	14
Charles De Gaulle	1.00	0.57	0.73	19
Nikita Khrushchev	1.00	0.38	0.55	13
Vincent Auriol	1.00	0.46	0.63	13
Konrad Adenauer	1.00	0.53	0.70	13
Dwight Eisenhower	0.85	0.46	0.60	13
Elisabeth II	1.00	0.71	0.83	14
Vyacheslav Molotov	1.00	0.23	0.37	13
Georges Pompidou	1.00	0.69	0.81	13
– unknown –	0.35	0.97	0.52	39
average (unknown apart)	0.97	0.59	0.71	216

Table 2. ANTRACT dataset: precision, recall, F-score and support for each class and aggregate results. The support column corresponds to the number of shots in which the person appears.

4.2 Quantitative analysis

For each dataset, a face recognition model has been trained to recognise the individuals from the corresponding list of celebrities. The model has then been applied to the video fragments of the ANTRACT and MeMAD datasets. We varied the confidence threshold t under which we considered the face not matched as shown in Figure 4, and found the optimal values with respect to the F-score – $t = 0.5$ for ANTRACT and $t = 0.6$ for MeMAD. The overall results – with the details of each person class – are reported in Table 2 and Table 3.

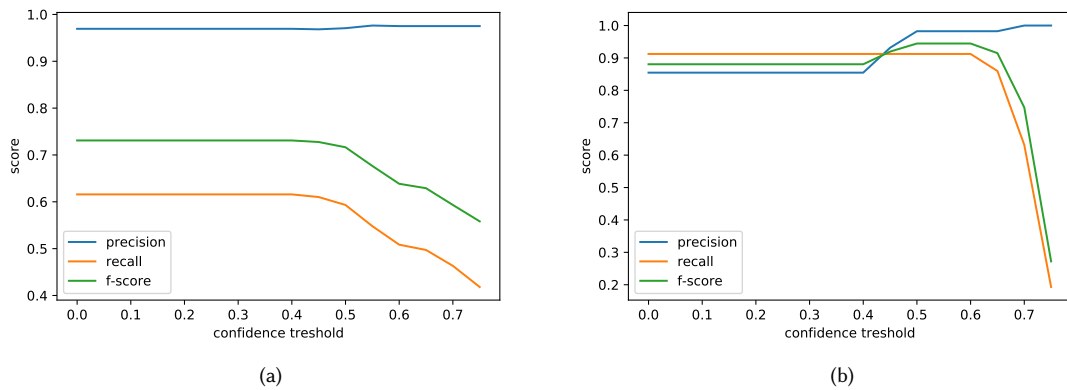


Fig. 4. Precision, recall and F-score of FaceRec on different confidence thresholds for the ANTRACT (4a) and the MeMAD dataset (4b).

The system obtained high precision in both datasets, with over 97% of correct predictions. If the recall on the MeMAD dataset is likewise good (0.91), it is significantly lower for the ANTRACT dataset (0.59). This is largely due to the differences between the two datasets, which involve not only the image quality, but also different shooting approaches.

Person	P	R	F	S
Le Saint, Sophie	0.90	0.90	0.90	10
Delahousse, Laurent	1.00	1.00	1.00	7
Lucet, Elise	1.00	0.90	0.94	10
Gastrin, Sophie	1.00	0.90	0.94	10
Rincquesen, Nathanaël de	1.00	0.80	0.88	10
Drucker, Marie	1.00	1.00	1.00	10
– unknown –	0.89	0.97	0.93	43
average (unknown apart)	0.98	0.91	0.94	100

Table 3. MeMAD dataset: precision, recall, F-score and support for each class and aggregate results. The support column corresponds to the number of segments in which the person appears.

If modern news are more used to close-up shots, taken on screen for multiple seconds, in historical videos, it is easier to find group pictures (in which occlusion is more probable), quick movements of the camera, and tight editing, leaving to our approach less samples for recognition. It is also relevant to notice that the lowest recall values belong to the only two USSR politicians Khrouchtchev and Molotov: most often, they appear in group images or in very short close-up images, raising questions for historical research.

4.3 Qualitative analysis

We made a qualitative analysis of the results. When inspecting the obtained recognition, we make the following observations:

- The system generally fails to detect people when they are in the background and their faces are therefore relatively small. This is particularly true for the ANTRACT dataset, in which the image quality of films is poorer.
- The cases in which one known person is confused with another known person are quite uncommon. Most errors occur when an unknown face is recognised as one of the known people.
- The recognition is negatively affected by occlusions of the face, such as unexpected glasses or other kind of objects.
- The used embeddings are not suitable to represent side faces, whose predictions are not reliable.

4.4 Unknown cluster detection evaluation

Together with the previous evaluation, we clustered the unknown faces found in the videos, as explained in Section 3.3. We then manually evaluated the resulting clusters on five randomly-selected videos for each dataset. We make the following observations:

- If more than one face is assigned to the same *Unknown <i>*, those faces actually belong to the same person. In other words, the erroneous presence of different individuals under the same label is never verified. This is due to the strict threshold chosen for intra-cluster distance.
- On the other side, not all the occurrences of that face are labelled, given that only the top five faces are kept. This may not be relevant if we are searching for new faces to add to the training set and we anyway intend to perform a further iteration afterwards.
- In one case, a single person was included in two distinct clusters, which may be reconciled by assigning the same label.

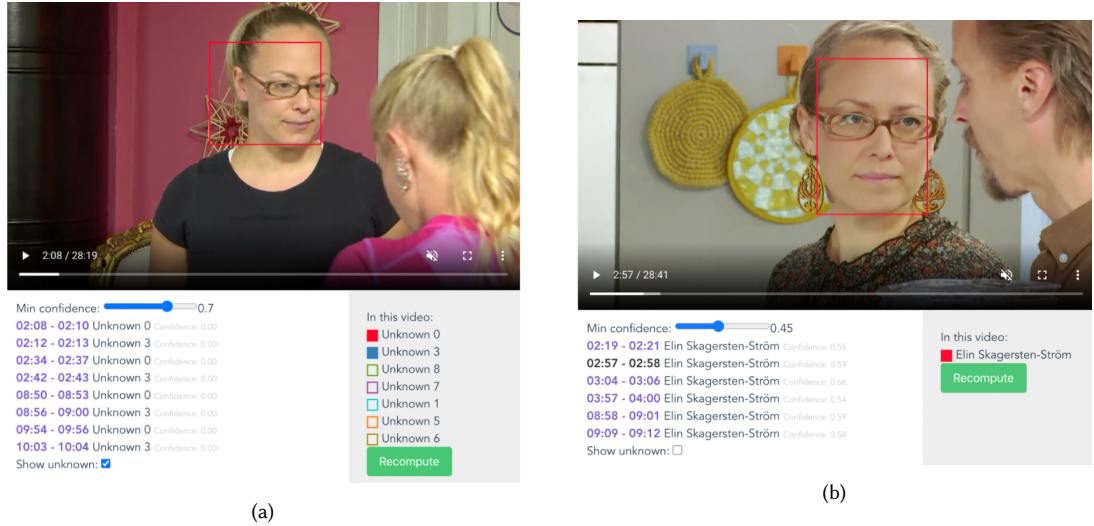


Fig. 5. The clustering output found a set of unknown persons in the video (5a). Using the frames of *Unknown 0*, we are able to build the model for Elin Skagersten-Ström and recognise her in other videos. (5b).

- Less clusters were found in the ANTRACT videos than in the MeMAD videos – three out of five videos with no clusters. This is again explained by the lower video quality, less frequent close-up shots and faster scene changes.

For understanding the benefit that results from the face clustering, we include in Figure 5 an example use case. In Figure 5a, the clustering algorithm identified a set of unknown people, among which *Unknown 0* happens to be Elin Skagersten-Ström, who was not part of our training set. For each segment in which *Unknown 0* appeared, we extracted the four frames closer to the middle of the segment and included them as images in the training set. By re-training the classifier with this new data, it was possible to correctly detect Elin Skagersten-Ström in other videos, as seen in Figure 5b. This approach can be applied to any individuals, including those for whom one cannot find enough face images on the Web for training a classifier.

5 A WEB API AND A USER INTERFACE

In order to make FaceRec publicly usable and testable, we wrapped its Python implementation within a Flask server and made it available as a **Web API** at <http://facerec.eurecom.fr/>. The API has been realised in compatibility with the OpenAPI specification¹⁴ and documented with the Swagger framework¹⁵. The main available methods are:

- `/crawler?q=NAME` for searching on the Web images of a specific person;
- `/train` for training the classifier;
- `/track?video=VIDEO_URI` for processing a video.

The results can be obtained in one of two output structures: a custom JSON format and a serialisation format in RDF using the Turtle syntax, relying on the EBU core¹⁶ and Web Annotation ontologies¹⁷. The Media Fragment URI¹⁸ syntax

¹⁴<https://www.openapis.org/>

¹⁵<https://swagger.io/>

¹⁶<https://www.ebu.ch/metadata/ontologies/ebucore/>

¹⁷<https://www.w3.org/ns/oa.ttl>

¹⁸<https://www.w3.org/TR/media-frags/>

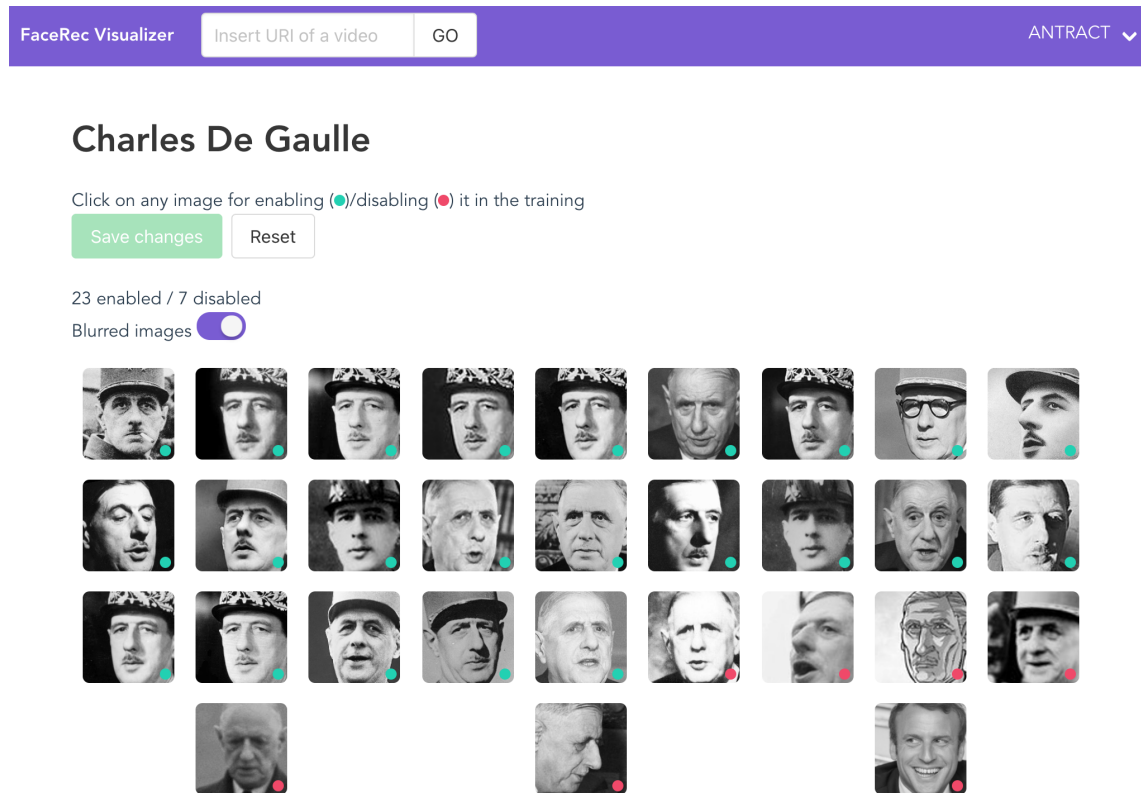


Fig. 6. Person page in FaceRec Visualizer: drawings, side faces, other depicted individuals and low-quality images are discarded (see the last 7 pictures marked with the red dot).

is also used for encoding the time and spatial information, with npt in seconds for identifying temporal fragments and $xywh$ for identifying the bounding box rectangle encompassing the face in the frame. A light cache system that enables to serve pre-computed results is also provided.

In addition, a **web application** for interacting with the system has been deployed at <http://facerec.eurecom.fr/visualizer>. The application has a homepage in which the list of celebrities in the training set is shown. For each person, it is possible to see the crawled images and decide which of them have to be included or excluded during the training phase (Figure 6). In addition, it is possible to add a new celebrity for triggering the automatic crawling and re-train the classifier once modifications have been completed.

Finally, it is possible to run the face recognition on a video, inserting its URI in the appropriate textbox. Partial results are shown to the user as soon as they are computed, so that it is not required to wait for the analysis of the entire video for seeing the first recognised faces. The detected persons are shown on a list, whose elements can be clicked for seeking the video until the relevant moment. The faces are identified in the video using squared boxes (Figure 5). A slider enables to vary the confidence threshold, allowing to interactively see the result depending on the value chosen. Some metadata are displayed for videos coming from the MeMAD and ANTRACT corpora.

6 CONCLUSIONS AND FUTURE WORK

With FaceRec, we managed to successfully exploit images on the web for training a face recognition pipeline which combines some of the best-performing state-of-the-art algorithms. The system has shown good performance, with an almost perfect precision. A clustering system has been integrated in FaceRec with unknown person detection, the results of which can be added to the training set. A web application allows to easily interact with the system and see the results on videos. The implementation is publicly available at <https://git.io/facerec> under an open source licence.

This system has been successfully applied in video summarisation, in a strategy combining face recognition, automatically-generated visual captions and textual analysis [11]. The proposed approach ranked first in the *TRECVID Video Summarization Task* (VSUM) in 2020.

In future work, we plan to improve the performances of our approach and in particular its recall. While the recognition of side faces largely impacts the final results, a proper strategy for handling them is required, also relying on relevant approaches from the literature [9, 18]. With quick changes of scenes, a face can be seen in the shot for only a very short time, not giving enough frames to the system for working properly. We may propose a different local sampling period $T_{local} < T$ to be used when a face is recognised in order to collect more frames close to the detection. In addition, we believe that the system would benefit from prior shot boundary detection in videos, in order to process shots separately.

A more solid confidence score can be returned including contextual and external information, such as metadata (the dates of the video and the birth-death of the searched person), the presence of other persons in the scene [15], and textual descriptions, captions and audio in multi-modal approaches [3, 10].

The presented work has several potential applications, from annotation and cataloguing to automatic captioning, with a possible inclusion in second-screen TV systems. Moreover, it can support future research in computer vision or in other fields – e.g. history studies. An interesting application is the study of age progression in face recognition [25].

ACKNOWLEDGMENTS

This work has been partially supported by the French National Research Agency (ANR) within the ANTRACT project (grant number ANR-17-CE38-0010) and by the European Union’s Horizon 2020 research and innovation program within the MeMAD project (grant agreement No. 780069).

REFERENCES

- [1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. 2006. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 28, 12 (2006), 2037–2041.
- [2] Adamu Ali-Gombe, Eyad Elyan, and Johan Zwigelaar. 2020. Towards a Reliable Face Recognition System. In *21st Engineering Applications of Neural Networks Conference (EANN)*, Lazaros Iliadis, Plamen Parvanov Angelov, Chrisina Jayne, and Elias Pimenidis (Eds.). Springer International Publishing, Cham, 304–316.
- [3] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. 2010. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems* 16, 6 (2010), 345–379.
- [4] Abdelkrim Beloued, Peter Stockinger, and Steffen Lalande. 2017. *Studio Campus AAR: A Semantic Platform for Analyzing and Publishing Audiovisual Corpora*. John Wiley & Sons, Ltd, Hoboken, NJ, USA, Chapter 4, 85–133.
- [5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple Online and Realtime Tracking. In *IEEE International Conference on Image Processing (ICIP)*. IEEE Computer Society, Phoenix, AZ, USA, 3464–3468.
- [6] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. 2018. VGGFace2: A Dataset for Recognising Faces across Pose and Age. In *13th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE Computer Society, Xi’an, China, 67–74.
- [7] Jean Carrire, Abdelkrim Beloued, Pascale Goetschel, Serge Heiden, Antoine Laurent, Pasquale Lisena, Franck Mazuet, Sylvain Meignier, Benedicte Pinchemin, Geraldine Poels, and Raphael Troncy. 2021. Transdisciplinary Analysis of a Corpus of French Newsreels: The ANTRACT Project. *Digital Humanities Quarterly, Special Issue on AudioVisual Data in DH* 15, 1 (2021).
- [8] Guodong Guo and Na Zhang. 2019. A survey on deep learning based face recognition. *Computer Vision and Image Understanding* 189 (2019).

- [9] Haroon Haider and Malik Khiyal. 2017. Side-View Face Detection using Automatic Landmarks. *Journal of Multidisciplinary Engineering Science Studies* 3 (2017), 1729–1736.
- [10] Anand Handa, Rashi Agarwal, and Narendra Kohli. 2016. A survey of face recognition techniques and comparative study of various bi-modal and multi-modal techniques. In *11th International Conference on Industrial and Information Systems (ICIIS)*. 274–279.
- [11] Ismail Harrando, Alison Reboud, Pasquale Lisena, Raphaël Troncy, Jorma Laaksonen, Anja Virkkunen, and Mikko Kurimo. 2020. Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization. In *International Workshop on Video Retrieval Evaluation (TRECVID 2020)*. NIST, Virtual Conference.
- [12] Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13, 2 (2002), 415–425.
- [13] Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the Special Issue on the Web as Corpus. *Computational Linguistics* 29, 3 (2003), 333–347.
- [14] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.
- [15] Yong Jae Lee and Kristen Grauman. 2011. Face Discovery with Social Context. In *British Machine Vision Conference (BMVA)*. BMVA Press.
- [16] Shan Li and Weihong Deng. 2020. Deep Facial Expression Recognition: A Survey. *IEEE Transactions on Affective Computing* (2020).
- [17] Hao Ma, Irwin Kink, and Michael R. Lyu. 2012. Mining Web Graphs for Recommendations. *IEEE Transactions on Knowledge and Data Engineering* 24 (2012), 1051–1064.
- [18] Pinar Santemiz, Luuk J. Spreeuwers, and Raymond N. J. Veldhuis. 2013. Automatic landmark detection and face recognition for side-view face images. In *International Conference of the BIOSIG Special Interest Group (BIOSIG)*. IEEE, Darmstadt, Germany.
- [19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Boston, MA, USA, 815–823.
- [20] Mohammad Shafin, Rojina Hansda, Ekta Pallavi, Deo Kumar, Sumanta Bhattacharyya, and Sanjeev Kumar. 2019. Partial Face Recognition: A Survey. In *3rd International Conference on Advanced Informatics for Computing Research (ICAICR)*. Association for Computing Machinery, Shimla, India, 1–6.
- [21] Paul Viola and Michael J. Jones. 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision* 57, 2 (2004), 137–154.
- [22] Howard Wactlar and Michael Christel. 2002. Digital Video Archives: Managing through Metadata. In *Building a National Strategy for Digital Preservation: Issues in Digital Media Archiving*. Library of Congress, Washington, DC, USA, 84–99.
- [23] Ivan William, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Heru Agus Santoso, and Christy Atika Sari. 2019. Face Recognition using FaceNet (Survey, Performance Test, and Comparison). In *4th International Conference on Informatics and Computing (ICIC)*. IEEE, Semarang, Indonesia.
- [24] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.
- [25] Huiling Zhou and Kin-Man Lam. 2018. Age-invariant face recognition based on identity inference from appearance age. *Pattern Recognition* 76 (2018), 191–202.

B.7 EURECOM's DataTV 2021b workshop paper

This paper describes the approach proposed by EURECOM for automatically segmented TV content and accepted at DataTV 2021 colocated with IMX 2021.

And cut!

Unsupervised Content Segmentation and Alignment

Ismail Harrando
ismail.harrando@eurecom.fr
EURECOM
Sophia Antipolis, France

Raphaël Troncy
raphael.troncy@eurecom.fr
EURECOM
Sophia Antipolis, France

ABSTRACT

Text segmentation is a traditional task in NLP where a document is broken down into smaller, coherent segments. While several methods and benchmarks exist for well-formed, and clean textual documents that can be found in long articles or synthetic datasets, segmenting media content comes with different challenges such as the errors produced by the procedure of Automatic Speech Recognition and the lack of sentence end markers that are found in written text (e.g. punctuation marks or HTML tags). Many radio or TV programs are also conversational in nature (e.g. interview and debate), and thus, rely less on repeated words unlike encyclopedia text (e.g. Wikipedia articles). This is even further compounded when working with non-English content. In this work, we present an unsupervised approach to content segmentation that leverages topical coherence, language modeling and word embeddings to detect change of topics. We evaluate our approach on real production data that has been manually annotated for segments.¹ We also show how, when a ground truth summary of the content is provided such as segment titles, we can align to their corresponding segments using the same representations.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**.

KEYWORDS

Content segmentation; Content Alignment; Unsupervised NLP

ACM Reference Format:

Ismail Harrando and Raphaël Troncy. 2021. And cut! Unsupervised Content Segmentation and Alignment. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

As the amount of multimedia content created and published every day has seen a remarkable growth in the recent years, the ability to serve end-users the content they are interested in becomes a

crucial ingredient to ensure their engagement. With their limited time and attention, users prefer content that is short and concise and topically relevant. There is therefore a need to segment available long-format content into shorter pieces. For instance, segmenting a news broadcast into multiple stories spanning different themes and topics can help online content distribution platforms to serve different users with different parts of the same broadcast. Content segmentation has also been shown to improve other media-related tasks such as content retrieval [16], content summarization [8], and sentiment analysis [9].

While the task of document or text segmentation has been studied extensively in the literature (Section 2), segmenting multimedia content present challenges that are particular to the medium: multimodality, automatic transcription errors, lack of proper punctuation, and presentation style (more informal talking, the use of pronouns and references instead of repeating words, etc.). To tackle the task of media content segmentation using automatically generated subtitles, we propose a textual approach that relies on combining several linguistic methods (topic modeling, words embeddings and sentence encoders) to generate richer representations of the content that we then use to predict segment boundaries.

2 RELATED WORK

While work on the task of document segmentation dates back to as least as early as 1984 [13], the most popular approach to text segmentation, *TextTiling*, was proposed by Hearst in 1997 [7], who devised an unsupervised approach in 3 steps: first, the text is divided into fixed-length sequences of words (called *blocks*), which are then transformed into a Bag of Words representation. The cosine similarity between adjacent sequences is computed, and the boundary between segments is determined at the position where the similarity is at its lowest, based on a sliding window. This classic text segmentation algorithm has been enhanced by different improvements addressing multiple challenges for the algorithm. [1] showed how introducing the time spoken by every participant in a recorded meeting as a feature can be used to better predict segment boundaries, as participants are typically not interested in every part of the meeting. [17] proposed to use word embeddings instead of word counts (bag of words) as more robust representations of the blocks to compare, and introduced a new heuristic to better capture the semantic coherence with the distributed document representations. More recently, He et al. [6] proposed an improvement over the last step of the algorithm, boundaries detection, by average-smoothing the similarity curve for adjacent blocks. This allows the local variations within topics to be smoothed-out whereas the topic switch would be perceived more clearly.

¹If you are interested in the dataset, please contact the authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DataTV-21, June 21–23, 2021, New York, NY

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

With the paradigm shift to neural networks in the 2010s, multiple neural models were proposed to address this task as a supervised learning problem. Recently, Lukasik et al. [11] proposed an approach based on BERT [5], where they compared 3 potential architectures to detect segment boundaries. They show that relying on attention between words and then between segments improves the results significantly on some standard benchmarks. Similarly, Yoong et al. [18] relied on BERT and the attention mechanism, and proposed 3 training pipelines: a naive approach with BERT is fed with two sentences and is trained to decide if they belong to the same segment or not; In a second approach, all sentences of the documents are fed to the model, and a decision is made on the [SEP] token separating them; finally, in a third approach, the segment boundary is modeled as a [SEP] token, and thus the task of segment prediction becomes one of a masked token prediction.

While the aforementioned methods are mostly evaluated on either synthetic datasets (where unrelated documents are concatenated to produce a segment boundary) or Wikipedia articles, some research work was particularly devoted to media content. In [15], Sehih et al. proposed a supervised approach based on a Bi-LSTM that is trained on a synthetically generated dataset to predict content segment of French News programs. Similarly, [14] propose an approach for automatic segmentation for Movie Subtitles to improve information retrieval from films. They based their approach on Text-Tiling, but used a synsets instead of a words to construct the Bag of Word representation of sentences. They also propose a filtering of segments based on the expectation that the similarity curve should be sinusoidal, and thus a minimum difference between the peaks (highest similarity) and valleys (lowest similarity) should be present to validate a proposed segment. [2] proposed improving automated segmentation of radio programs by adding audio embeddings to the text input. [19] used a temporal convolutional network (TCN) combined with BERT features to perform dialog stream segmentation, while introducing speaker information as part of the input sequence, and they observe significant improvement over several dialog segmentation datasets.

3 APPROACH

The main steps of our approach are similar to TextTiling [7], i.e. partitioning text fixed-size sequences of words, or *blocks*, computing pairwise similarity between adjacent blocks, and assigning segment ends to the minima of the similarity curve (Figure 3). We extend this approach by leveraging on multiple text representations instead of simple word counts, and smoothing the similarity curve by considering a window of adjacent similarity.

The high-level description of our approach is illustrated in Figure 1. We detail each steps in the following subsections.

3.1 Transcript Partitioning

One of the main differences between traditional documents and automatically generated transcripts is the lack of natural sentence end markers. While most ASR systems cut long utterances into smaller sentences, they vary considerably in length, and tend to be too short to carry meaningful topical information. As a simple partitioning method, we divide the content of each program, as

generated by the ASR system and after removing stopwords, into *blocks* of a fixed number of words per block N .

3.2 Text Representation

To find segment boundaries, we need to find the blocks in the transcript where a *topic shift* takes place, i.e. where the similarity between the current block and the following one (or ones), is lowest. To do that, we generate several textual representations that allow us to measure similarity between blocks from the transcript. Since all these methods produce a fixed-size vector representation, we compute the similarity between blocks using cosine similarity (i.e. normalized dot product).

The curve of adjacent blocks similarity tends to be spiky: a lot of peaks and valleys come naturally from the variability of the vocabulary between immediately consecutive utterances. Therefore, we also consider measuring the similarity of each block to the ones following it within a perimeter of *window_size*. This has both a smoothing effect for sharp transitions in similarity as well as removing saddle points (stretches of the curve where the score does not change).

3.2.1 Word Embeddings. Pretrained word embeddings have been a fixture in most NLP tasks, especially for unsupervised methods. For our experiments, we use the pretrained French *fastText* embeddings [3]. Beyond their empirical performance as standalone word embeddings, fastText embeddings have the capacity of generating a representation even for words that are outside of their training vocabulary by leveraging their sub-word components. We use the 300-d pretrained vectors, available on their official website².

3.2.2 Sentence Encoder. Another way to represent the textual content is through the use of Sentence Encoders which attempt to capture the meaning of a sentence through both its constituent words and its grammatical structure. While there is a rich literature on the topic, most state-of-the-art applications use *Sentence-BERT* [12], which uses pretrained BERT to construct semantically meaningful sentence embeddings that can be compared using cosine-similarity. We use the *sentence-transformers* Python package³ to generate sentence embeddings for our program content.

3.2.3 Topic Modeling. Since the ultimate goal of this task is to segment text into topically coherent segments, it shares several aspects with Topic Modeling. While generally used to infer topic information about given texts, the output of a topic model can be used as a "feature vector", or a representation of a given text, i.e. as a linear combination of its latent topical components. We select LDA as our topic model based on empirical evaluation of several models (using the Python library *Tomodapi* [10]). We train the model on a synthetic dataset that we create by concatenating adjacent blocks from our original dataset (as adjacent blocks are highly likely to talk about the same topic) as well as succeeding lines from the automatically generated transcript (i.e. before partitioning into blocks). It is worth noting that LDA has also the property of producing sparse representation, i.e. every document only falls into a few (3 or less) topics, which makes most of the representation components null.

Figure 2 visualizes the representations for an example on the dataset using LDA features.

²<https://fasttext.cc/docs/en/crawl-vectors.html>

³<https://github.com/UKPLab/sentence-transformers>

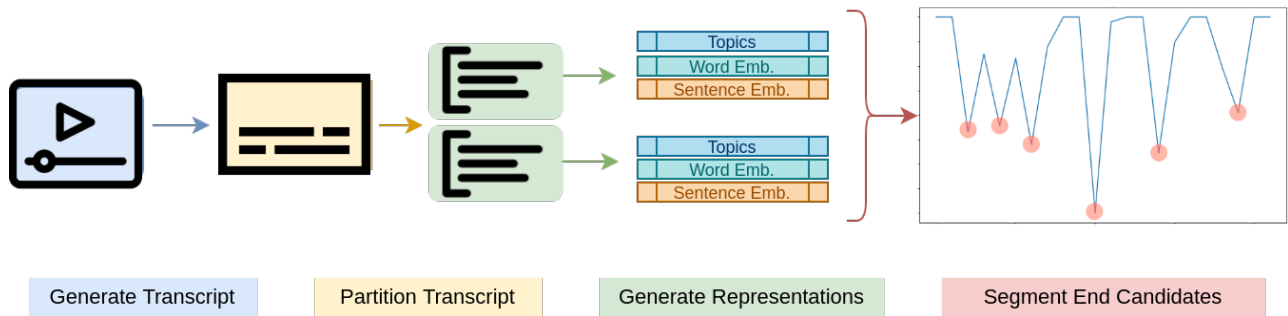


Figure 1: High level illustration of the approach: (1) Generate a transcript of the program using ASR. (2) Partition the transcript into *blocks* of equal size N . (3) Generate different representations of the textual content of each block. (4) After measuring the similarity between each block and its neighborhood, each "valley" in the similarity curve is a candidate to be the topic transition block (i.e. end of the segment)

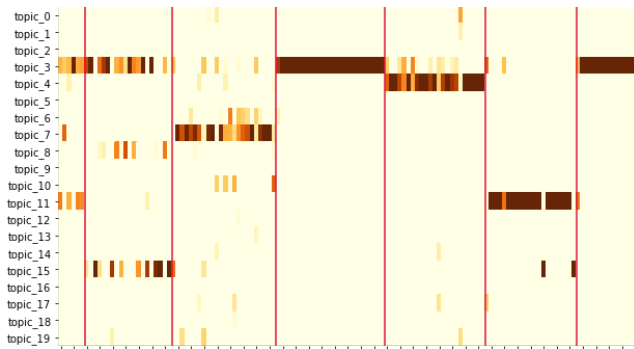


Figure 2: Visualizing the topic distribution over an example in the dataset. The vertical lines represent the ground truth segment boundaries

3.3 Boundaries selection

As mentioned above, we consider a *boundary candidate* to be a minima in the similarity curve, i.e. the similarity scores resulting from comparing the content of the block at position i with that at position $i + 1$. In the case of $window_size > 1$, we average the similarity scores between the content at block i and all blocks between $i + 1$ and $i + window_size$. Figure 3 shows an example of the process.

An important parameter in the boundaries selection is the *number of segments* in the program. For our model selection and comparison, we consider the number of parts as given, i.e. for every program, we only propose as many segments as there are in the ground truth. We show in Section 4 some simple heuristics to replace this ground truth information.

4 EXPERIMENTS AND RESULTS

In this section, we describe the dataset we are using for our experiments as well as the different experimental settings. For the evaluation, we consider segmentation as a classification task, where each block is assigned a label: 0 if it is part of a homogeneous segment, or 1 if it represents a topic transition block, i.e. having a low similarity to the blocks following it.

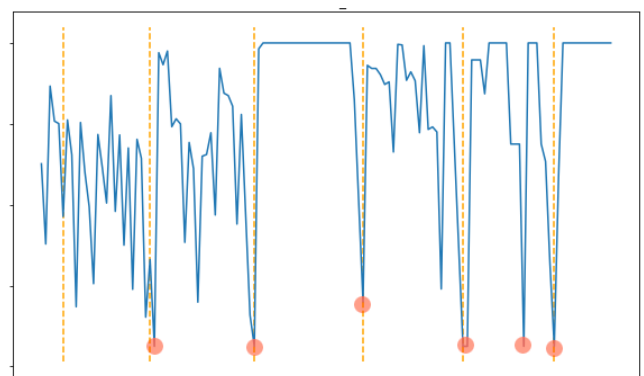


Figure 3: An example of a similarity curve generated by topical similarity. The circles highlight the valleys that correspond to the segment boundaries selected by our approach. We note that in this case, the number of segments is given. The dashed lines represent the ground truth segment boundaries

4.1 Dataset

For our evaluation, we use a production dataset from INA (the French National Audiovisual Institute) containing 46 programs from the same week of publication (May 19th to 26th, 2014), with a total runtime of 15 hours, that were segmented into 476 parts, of 112 seconds duration in average. The segmentation is done manually by archivists and each part is given a title. Most of the programs that are provided are news broadcasts, with the segments corresponding to news stories, but the dataset also includes some sport and cultural event coverage. Each program in this dataset has been automatically transcribed using the LIUM ASR System [4]. It is worth noting that the segmentation boundaries contain some noise as they do not perfectly align with ASR nor usually add up to the total duration of the program.

4.2 Segmentation

For each of the textual representation, we evaluate the data using traditional classification measures (Precision, Recall, F1 score) which quantify the amount of exact segment boundaries detected

by each method, as well as two segmentation-specific metrics: P_k and $WindowDiff_k$. These metrics represent the probability that two blocks are wrongly assigned to the same segment, but tolerating some error within k blocks of the ground truth (close matches), and differ in how each penalizes said close matches. We set $k=2$ as it is conventionally set to half the average length of a segment. We note that for P_k and $WindowDiff_k$, the closer their score is to 0, the better is the segmentation.

Considering the three text representations described in Section 3, we propose several variants:

- **Sentence-BERT**: we consider three variants representing pre-trained multilingual models on different tasks: `distiluse-base-multilingual` (distilled base multilingual BERT), `paraphrase-xlm-r-multilingual` (XLM fine-tuned on the task of paraphrasing), and `stsb-xlm-r-multilingual` (XLM fine-tuned on the task of Semantic Textual Similarity Benchmark).
- **fastText**: for both variants we use pretrained French fastText embeddings. We test two similarity measures: averaging all the embeddings in each block to form a block representation that is then used for cosine similarity (`fastText-avg`), or, as suggested by [17], we keep the best cosine similarity between two blocks, i.e. the similarity scores for the most similar words in the two successive blocks (`fastText-max`).
- **LDA**: We train an LDA model with the same hyper parameters with different number of topics, thus changing the size of the representation vector. We set both α and η to ‘auto’, while varying the number of topics between 10, 20 and 30.

As previously mentioned, the similarity scores are computed using cosine similarity (normalized dot product) between the vector representations of adjacent blocks or within a window thereof. We set the block size $N=20$.

In Table 1, we show the results on the INA dataset using the different text representations used to measure textual similarity between content blocks. For the *Combined* line, we try a linear combination of similarity scores generated from the best performing variant from each representation, and we empirically select the combination (0.6, 0.3, 0.1) for LDA-20, `fastText-avg` and S-BERT-`paraphrase`, respectively. Among the text representations, we see that LDA performs best for both the classification and segmentation metrics. The combined score, however, significantly outperforms the individual representations, showing that each of the representations contain different but complementary information.

In Table 2, we improve on the previous approach by extending the similarity measure to a window of size > 1 , as the smoothing effect can cover some of the noise that is present in the data. This turns out to be the case, as extending the similarity to a vicinity of 3 (selected empirically) blocks instead of just one, we see a noticeable improvement over almost all representations. We also report the best results on the *Combined* representation, which still significantly outperforms all existing methods.

4.2.1 Block Size. In this section, we study the empirical effect of the size of the unit partitioning block. We repeat the experiments explained in this section for block size 10, 20 and 30. In Table 3, we report the results on the dataset using the *Combined* representation

Approach	Pre	Rec	F1	P_k	WD
S-BERT-paraphrase	0.235	0.311	0.261	0.467	0.505
S-BERT-distiluse	0.255	0.343	0.284	0.445	0.476
S-BERT-stsb	0.266	0.352	0.296	0.447	0.495
fastText-max	0.235	0.271	0.251	0.416	0.440
fastText-avg	0.258	0.300	0.277	0.401	0.439
LDA ($N=10$)	0.297	0.377	0.330	0.378	0.424
LDA ($N=20$)	0.291	0.421	0.335	0.398	0.447
LDA ($N=30$)	0.297	0.440	0.344	0.412	0.474
Combined	0.321	0.371	0.344	0.355	0.392

Table 1: Segmentation results on the INA dataset ($window_size=1$)

Approach	Pre	Rec	F1	P_k	WD
TextTiling	xx	xx	xx	xx	xx
S-BERT-paraphrase	0.281	0.377	0.313	0.427	0.492
S-BERT-distiluse	0.253	0.342	0.283	0.443	0.503
S-BERT-stsb	0.270	0.352	0.298	0.422	0.474
fastText-max	0.245	0.281	0.262	0.423	0.451
fastText-avg	0.278	0.324	0.298	0.399	0.454
LDA ($N=10$)	0.397	0.469	0.429	0.313	0.368
LDA ($N=20$)	0.399	0.473	0.431	0.319	0.370
LDA ($N=30$)	0.374	0.453	0.409	0.340	0.396
Combined	0.431	0.500	0.462	0.291	0.345

Table 2: Segmentation results on the INA dataset ($window_size=3$)

with $window_size=3$, as it still performs best among all proposed approaches.

Block Size	Pre	Rec	F1	P_k	WD
10	0.178	0.327	0.222	0.320	0.334
20	0.431	0.500	0.462	0.291	0.345
30	0.521	0.345	0.400	0.419	0.456

Table 3: Comparing performance as a function of the partitioning block size

From the results, we see clearly that for $N=10$, the smaller blocks fail to capture enough topical information, as we see a significant drop in all metrics. As for $N=30$, we see an increase of Precision (i.e. a higher ratio of true positives), but at the cost of recall and overall F1-score.

4.2.2 Number of segments. For the previous experiments, we set the number of segments for each program to be equal that of the ground-truth, which is an ideal setting just to evaluate the performance of the representations. In Table 4, we experiments with two simple heuristics:

- **1/6**: we pick the number of segments to be equal to a sixth of the number of blocks generated for the program. As we computed the ratio of *blocks to segment* to be equal to 16.
- **Thresh**: we only keep the segmentation candidate at position i if it satisfies the following inequality:

$$hi - minhri, hli$$

$$\frac{1}{N} \sum_j h_j - sim_j, j - hi - sim_i, i - 1 < 0$$

with hri and hli two functions returning the highest peak to the right and the left of i , respectively, and they are both defined for each program. In concrete terms, this means we only keep the candidates which are situated at valleys that are deeper than the average valley in the entire similarity curve.

- **GT** (ideal case): we reproduce the results from the previous experiments with the number of segments to be picked is equal to that of the ground truth.

Block Size	Pre	Rec	F1	P_k	WD
GT	0.431	0.500	0.462	0.291	0.345
1/6	0.266	0.478	0.340	0.278	0.297
Thresh	0.451	0.297	0.384	0.329	0.394

Table 4: Comparing performance as a function of the number of segment selection method

As we see the results in Table 4, the different methods offer different compromises. While using 16, by virtue of detecting less boundaries on short programs, we get better P_k and $WindowDiff_k$ scores than when using the ground truth, but the classification scores are comparatively low. Whereas for *Thresh*, we get segmentation scores that are close to GT, while not losing as much in classification scores.

4.3 Aligning segments with description metadata

In our ground truth, every annotated segment is given a title that corresponds to a summary of its content. Given how in production there is typically metadata about the content of the program (e.g. segment titles), we further test the scenario of aligning the automatically generated transcript with the existing metadata. In this setting, we consider the number of segments given (to be equal to the number of provided segment titles), and we create an alignment by measuring the similarity between each block in the transcript (we keep the block size $N = 20$) and a title from the ground truth annotation, using all the representations we mentioned above. Starting from the first title, we set a segment boundary at each position where a block has more similarity to the next title than the one currently considered.

As we can see in Table 5, the results based on content alignment with the titles, while comparable to the segmentation results on P_k and $WindowDiff$, are significantly lower on classification metrics. Upon analysis, we see that this is probably due to the shortness of the descriptive titles, which do not carry enough information to measure similarity significantly, regardless of the chosen textual representation (all methods perform comparatively the same). A combined decision (obtained by assigning the coefficients 0.5, 0.3, 0.2 to the

Approach	Pre	Rec	F1	P_k	WD
S-BERT-paraphrase	0.281	0.377	0.313	0.427	0.492
fastText-avg	0.241	0.243	0.243	0.406	0.448
LDA ($N = 20$)	0.264	0.263	0.264	0.387	0.432
Combined	0.390	0.271	0.319	0.296	0.342

Table 5: Alignment results on the INA dataset

similarity score of S-BERT, fastText and LDA, respectively), however, does improve the results, which highlights again the fact that leveraging on multiple textual representations is key to improving the overall segmentation results.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a new method for unsupervised content segmentation based on combining multiple text representations, and we show that topic modeling is a useful representation. More advanced methods for deriving and combining the representations, as well as finding the number of segments in the program, can be considered in the future. We would also like to explore the use of multimodal features to further improve the segmentation: audio features such as silence periods and speaker turns, and visual features (e.g. visual shot similarity, scene segmentation) can also help complementing textual content for programs that present more visual diversity.

6 ACKNOWLEDGMENTS

This work has been partially supported by the French National Research Agency (ANR) within the ANTRACT (ANR-17-CE38-0010) project, and by the European Union’s Horizon 2020 research and innovation program within the MeMAD (GA 780069) project.

REFERENCES

- [1] Satantjeet Banerjee and Alexander I. Rudnicky. 2006. A texttiling based approach to topic boundary detection in meetings. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, September 17-21, 2006. ISCA. http://www.isca-speech.org/archive/interspeech_2006/i06_1827.html
- [2] Oberon Berlage, Klaus-Michael Lux, and David Graus. 2020. Improving Automated Segmentation of Radio Shows with Audio Embeddings. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2020). DOI:<http://dx.doi.org/10.1109/icassp40776.2020.9054315>
- [3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. DOI:http://dx.doi.org/10.1162/tac1_a_00051
- [4] Fethi Bougares, Paul Deléglise, Yannick Estève, and Mickael Rouvier. 2013. LIUM ASR system for ETAPE French evaluation campaign: Experiments on system combination using open-source recognizers, Vol. 8082. 319–326. DOI:http://dx.doi.org/10.1007/978-3-642-40585-3_41
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. DOI:<http://dx.doi.org/10.18653/v1/n19-1423>
- [6] Xin He, Jian Wang, Quan Zhang, and Xiaoming Ju. 2020. Improvement of Text Segmentation TextTiling Algorithm. *Journal of Physics: Conference Series* 1453 (jan 2020), 012008. DOI:<http://dx.doi.org/10.1088/1742-6596/1453/1/012008>

- [7] Marti A. Hearst. 1997. TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. *Comput. Linguist.* 23, 1 (March 1997), 33–64.
- [8] Joel Larocca Neto, Alexandre D. Santos, Celso A.A. Kaestner, and Alex A. Freitas. 2000. Generating Text Summaries through the Relative Importance of Topics. In *Advances in Artificial Intelligence*, Maria Carolina Monard and Jaime Simão Sichman (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 300–309.
- [9] J. Li, B. Chiu, S. Shang, and L. Shao. 2020. Neural Text Segmentation and Its Application to Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1. DOI:<http://dx.doi.org/10.1109/TKDE.2020.2983360>
- [10] Pasquale Lisena, Ismail Harrando, Oussama Kandakji, and Raphael Troncy. 2020. TOMODAPI: A Topic Modeling API to Train, Use and Compare Topic Models. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics, Online, 132–140. DOI:<http://dx.doi.org/10.18653/v1/2020.nlp-oss-1.19>
- [11] Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. 2020. Text Segmentation by Cross Segment Attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4707–4716. DOI:<http://dx.doi.org/10.18653/v1/2020.emnlp-main.380>
- [12] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. DOI:<http://dx.doi.org/10.18653/v1/D19-1410>
- [13] John A. Rotondo. 1984. Clustering analysis of subjective partitions of text. *Discourse Processes* 7, 1 (1984), 69–88. DOI:<http://dx.doi.org/10.1080/01638538409544582>
- [14] Martin Scaiano, Diana Inkpen, Robert Laganière, and Adele Reinhartz. 2010. Automatic Text Segmentation for Movie Subtitles. In *Advances in Artificial Intelligence*, Atefeh Farzindar and Vlado Kešelj (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 295–298.
- [15] Imran A. Sheikh, Dominique Fohr, and Irina Illina. 2017. Topic segmentation in ASR transcripts using bidirectional RNNs for change detection. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*. IEEE, 512–518. DOI:<http://dx.doi.org/10.1109/ASRU.2017.8268979>
- [16] Gennady Shtekh, Polina Kazakova, Nikita Nikitinsky, and Nikolay Skachkov. 2018. Applying Topic Segmentation to Document-Level Information Retrieval. In *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia (CEE-SECR '18)*. Association for Computing Machinery, New York, NY, USA, Article 6, 6 pages. DOI:<http://dx.doi.org/10.1145/3290621.3290630>
- [17] Yiping Song, Lili Mou, Rui Yan, Li Yi, Zinan Zhu, Xiaohua Hu, and Ming Zhang. 2016. Dialogue Session Segmentation by Embedding-Enhanced TextTiling. *CoRR* abs/1610.03955 (2016). <http://arxiv.org/abs/1610.03955>
- [18] Siang Yun Yoong, Yao-Chung Fan, and Fang-Yie Leu. 2021. On Text Tiling for Documents: A Neural-Network Approach. In *Advances on Broad-Band Wireless Computing, Communication and Applications*, Leonard Barolli, Makoto Takizawa, Tomoya Enokido, Hsing-Chung Chen, and Keita Matsuo (Eds.). Springer International Publishing, Cham, 265–274.
- [19] L. Zhang and Q. Zhou. 2019. Topic Segmentation for Dialogue Stream. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 1036–1043. DOI:<http://dx.doi.org/10.1109/APSIPAASC47483.2019.9023126>

B.8 EURECOM's DHQ 2021 journal paper

This paper describes extensive experiments made by EURECOM using the FaceRec library in analyzing old newsreel from the Partner INA and published in the Digital Humanities Quarterly journal, volume 15, number 1.

Transdisciplinary Analysis of a Corpus of French Newsreels: The ANTRACT Project

Jean Carrive, Abdelkrim Beloued, Pascale Goetschel, Serge Heiden, Antoine Laurent, Pasquale Lisena, Franck Mazuet, Sylvain Meignier, Bénédicte Pincemin, Géraldine Poels, et al.

► To cite this version:

Jean Carrive, Abdelkrim Beloued, Pascale Goetschel, Serge Heiden, Antoine Laurent, et al.. Transdisciplinary Analysis of a Corpus of French Newsreels: The ANTRACT Project. Digital Humanities Quarterly, Alliance of Digital Humanities, 2021, Special Issue on AudioVisual Data in DH, 15 (1). hal-03166755

HAL Id: hal-03166755

<https://hal.archives-ouvertes.fr/hal-03166755>

Submitted on 11 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NoDerivatives| 4.0 International License

Transdisciplinary Analysis of a Corpus of French Newsreels: The ANTRACT Project

JEAN CARRIVE¹, ABDELKRIM BELOUED¹, PASCALE GOETSCHER², SERGE HEIDEN³, ANTOINE LAURENT⁴, PASQUALE LISENA⁵, FRANCK MAZUET², SYLVAIN MEIGNIER⁴, BÉNÉDICTE PINCEMIN³,
GÉRALDINE POELS¹, RAPHAËL TRONCY⁵

¹ INA – Institut national de l’audiovisuel

4 avenue de l’Europe, 94366 Bry-sur-Marne Cedex, France

² Centre d’histoire sociale des mondes contemporains, UMR 8058 (Université Paris 1/CNRS)
Campus Condorcet, bâtiment recherche sud, 5 cours des Humanités, 93300 Aubervilliers, France

³ Univ. Lyon, IHRIM, Institut d’histoire des représentations et des idées dans les modernités,
UMR 5317

ENS de Lyon, 15 parvis René Descartes, BP 7000 69342 Lyon Cedex 07, France

⁴ LIUM, Laboratoire d’Informatique de l’Université du Mans
Avenue Olivier Messiaen F-72085 – Le Mans Cedex 9, France

⁵ EURECOM, 450 Route des Chappes, 06410 Biot, France

¹jcarrive at ina.fr, abeloued at ina.fr, gpoels at ina.fr

²fmazuet at free.fr, pascalle.goetschel at univ-paris1.fr

³slh at ens-lyon.fr, benedict.e.pincemin at ens-lyon.fr

⁴sylvain.meignier at univ-lemans.fr, antoine.laurent at univ-lemans.fr

⁵pasquale.lisena at eurecom.fr, raphael.troncy at eurecom.fr

Abstract

The ANTRACT project is a cross-disciplinary apparatus dedicated to the analysis of the French newsreel company *Les Actualités Françaises* (1945-1969) and its film productions. Founded during the liberation of France, this state-owned company filmed more than 20,000 news reports shown in French cinemas and throughout the world over its 24 years of activity. The project brings together research organizations with a dual historical and technological perspective. ANTRACT’s goal is to study the production process, the film content, the way historical events are represented and the audience reception of *Les Actualités Françaises* newsreels using innovative AI-based data processing tools developed by partners specialized in image, audio, and text analysis.

This article focuses on the data processing apparatus and tools of the project. Automatic content analysis is used to select data, to segment video units and typescript images, and to align them with their archival description. Automatic speech recognition provides a textual representation and natural language processing can extract named entities from the voice-over recording; automatic visual analysis is applied to detect and recognize faces of well-known characters in videos. These multifaceted data can then be queried and explored with the TXM text-mining platform.

The results of these automatic analysis processes are feeding the Okapi platform, a client-server software that integrates documentation, information retrieval, and hypermedia capabilities within a single environment based on the Semantic Web standards. The complete corpus of *Les Actualités Françaises*, enriched with data and metadata, will be made available to the scientific community by the end of the project.

1. Implementing a Transdisciplinary Research Apparatus on a Film Archive Collection: Opportunities and Challenges

The ANTRACT¹ project brings together research organizations with a dual historical and technological perspective, hence the reference to the transdisciplinary in the project's name. It applies to a collection of 1262 newsreels (mostly black and white footage) shown in French movie theaters between 1945 and 1969. These programs were produced by *Les Actualités Françaises* newsreel company during the French *Trente Glorieuses* era. The project develops automated tools well suited to analyze these documents: automatic speech recognition, image classification, facial recognition, natural language processing, and text mining. These software are used to produce metadata and to help organize media files and documentation resources (i.e. titles, summaries, keywords, participants, etc.) into a manageable and coherent corpus usable within a dedicated online platform.

Working together on these newsreels divided into 20,232 news reports, ANTRACT historians and computer scientists collaborate to optimize the research on large audiovisual corpora through the following questions:

- What is the best technological approach to the systematic and exhaustive study of a multimedia archive collection?
- What instruments can compile, analyze and crosscheck the data extracted from such documents?
- Can these extracted data be combined and integrated into versatile user interfaces?
- Can they provide new opportunities to humanities research projects through their assistance in the processing of numerous multi-format sources?

In order to implement a strong cooperation between AI experts and history scholars (Deegan and McCarty, 2012), the key objective of the project is to provide scholars and media professionals working on extensive collections of film archives with an innovative research methodology fit to address the technological and historical questions raised by this particular corpus.

From a technological perspective, the goal is to adapt automatic analysis tools to the specificity of the *Actualités Françaises* corpus, i.e. its historical context, vocabulary, image type. Adapting the language models used by the automatic transcription tools with the help of the typescripts of voice overs underlines this orientation. As a film collection including footage, sound and text produced more than half a century ago, *Les Actualités Françaises* corpus presents an unprecedented challenge to instruments specialized in audiovisual content extraction and identification. Far from separately considering a social and cultural history of cinema on the one hand, and the use of automatic analysis tools on the other hand, the project aims to link the two. Thus, a good understanding of the technical conditions for recording the audio leads to improved audio recognition. Shot in black and white with limited equipment and often under difficult filming conditions, these old newsreels do not meet the quality standards set by the high definition video and audio recordings feeding today's image and speech recognition algorithms. Moreover, several film reels of the collection digitized under high compression formats show pixelated images that cannot be processed by analysis programs and some of the commentary typescripts display printing defects caused by the typewriters used for their production.

Along with these material obstacles comes the problem raised by the transfiguration of film content over time. This is the case for leading figures regularly filmed by the company's cameramen throughout its 24 years of activity. It is also the case for the recurring topographical data caught on

their film. The automatic identification of these ever-changing elements recorded on monochromatic footage requires a considerable amount of resources. As part of this process, ANTRACT historians have selected a sample of the most distinctive representations of notable characters present in *Les Actualités Françaises* newsreels in order to build a series of extraction models.

From an historical perspective, ANTRACT aims to approach topics beyond the notion of newsreels as a wartime media subjected to state censorship and political ambitions (Atkinson, 2011; Bartels, 2004; Pozner, 2008; Veray, 1995). In the wake of existing studies, one of its primary objectives is to extend the historical scope of the cinematographic press to question its role as a vector of social, political and cultural history shaping the opinion of the public during the second half of the 20th century (Fein, 2004, 2008; Althaus et al., 2018; Chambers et al., 2018; Imesch et al., 2016; Lindeperg 2000, 2008). This series of cinematographic documents is not the only legacy left by a newsreel company which witnessed world history from the liberation of France to the late 1960's. The dope sheets filled out by its cameramen, the written commentaries of its journalists and the records left by its management give us rare insight into the content of a film collection as well as its production process. Despite its historical value, *Les Actualités Françaises* corpus has eluded a thorough examination of its entire content. Scattered across different inventories, the numerous films, audio records and typescripts produced by the newsreel company have forestalled such a project. In this regard, the challenge presented by an exhaustive study of *Les Actualités Françaises* is similar to those of other abundant multi-format collections and inspires a recurring question regarding their approach: how can one identify and index thousands of hours of film archives associated with hundreds of text files produced over an extended period of time? The tools developed by the consortium partners working on the project are intended to cast a new light on the French company newsreels through the combined treatment of data extracted from its whole collection and correlatively studied on the Okapi and TXM platforms. This apparatus should open new semantic fields previously overlooked by the fragmentary research conducted on specific inventories of the company records. Focused on film content, the project is also committed to scrutinize the production process and the different trades involved in the making of *Les Actualités Françaises* newsreels emphasizing the political and economic background of a company controlled by a democratic state. Underlining the notion that media participate in events (Goetschel and Granger, 2011), this dual analysis - both technological and historical - will be completed with the study of the public reception of these weekly journals in light of its request patterns, i.e. audience expectation for sensational and exotic news and its interest in the daily life of renowned figures (Maitland, 2015).

Through audio and video analysis tools dedicated to corpus building and enrichment (section 2) and platforms for historical interactive analysis (section 3), this article presents the results from the first phase of the project, which sets the focus on the technological side of the research, specifically its data processing apparatus and tools. Nevertheless, historians are involved in most of these computational preliminary steps, by contributing to the implementation and testing tasks. At the same time, we explore temporary results of historical investigations, while the full potential for historical studies will be developed in the forthcoming second phase of the project that will be addressed in a follow-up article.

2. Corpus building and enrichment

2.1 Organization of the video corpus with automatic content analysis technologies

Automatic content analysis technologies are used to obtain the most consistent, complete and homogeneous corpus as possible, allowing historians to easily search and navigate through the documents (digitized films, documentation notes and typescripts). When considering that the whole archive would not be relevant, a preliminary step was to realize that for some tasks, we had to define how our corpus would be composed and structured. One cannot just input the data into the computer and see what happens. For instance, textometric analysis would be hindered if all the available videos were kept, because of numerous duplicates which would artificially inflate word frequencies. Duplicates could be due to either multiple copies of a single news report, or to the use of the same report in several regional editions. As a collaborative decision involving newsreel experts, corpus analysis researchers, and historians, ANTRACT's main corpus was restricted to the collection of all national issues of *Les Actualités Françaises* newsreels, each issue being composed of topical report sub-units. Then, the next goal was:

- 1) to get a corpus made of exactly one digital video file by edition (which was a requisite condition for TXM data import, see Section 3.1),
- 2) to get archival descriptions of the reports temporally linked to these files, as an edition is made of a succession of reports.

This led us to take the following actions:

- 1) physically segment video files initially coming from the digitization of film reels, so that each file contains exactly one edition, starting at timecode 0.
- 2) keep only archival descriptions linked to either one edition or one report included in one of the editions, namely “summary” and “report” archival descriptions. Thus, archival descriptions corresponding to other content, such as rushes or unused material, called “isolated” archival descriptions were discarded. Around 10,700 archival descriptions have thus been kept in this first version of the corpus.

The remaining of this section explains how automatic analysis has been used to temporally synchronize archival descriptions with digital video files.

Segmentation of reports. Each one of the 1,200 editions of the newsreels corresponds to more than one digital video file, either because several digitized copies of one given edition exist in the collection, or because the film has been digitized several times, for quality reasons for instance. When they exist, timecodes of archival descriptions may refer to one or the other digital video file. One objective is to get all archival descriptions of one edition referring to the same video file, with timecodes. About 9,500 out of 10,700 archival descriptions have timecodes referring to the video file of the whole edition, which left around 1,200 archival descriptions to manage. A report with timecode is called “*segmented*”. One important step is to segment each edition into its constitutive reports, by detecting report boundaries. In most cases, reports are separated by black images, easily detected by simple image analysis methods (the *ffmpeg* video library offers an efficient “blackdetect” option for instance). Reports may also be separated by sequences of a few frames to a few seconds of a motion blur shot by a camera, used as a syntactic punctuation. In some cases, when these sequences are long enough, they can be detected as a simple threshold on the horizontal dimension of the optical flow,

computed with existing algorithms such as OpenCV (Bradski, 2000). A more robust detection method is still under development using machine learning algorithms.

Transfer of timecodes. When timecodes refer to a video file different from the main video file, timecodes on the main file may be computed using copy detection techniques. The principle is illustrated by Figure 1. In the figure, reports on “Rugby” and “Kennedy’s visit” (from the edition of May 31st, 1961) refer to two video files, both distincts from the video file corresponding to the whole edition. To identify the location of the reports within the main video file, we used the audio and video copy detection method based on fingerprinting methods developed at INA (Chenot and Daigneault, 2014), eventually allowing the transfer of timecodes for more than 800 reports.

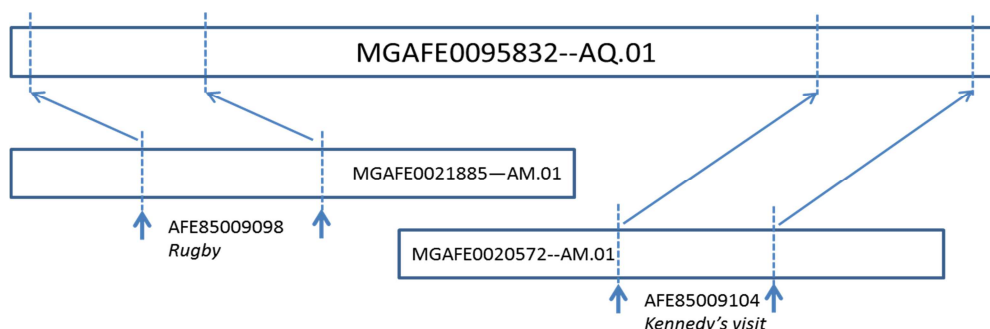


Figure 1. Transfer of archival description timecodes

Timecoding reports using transcripts. We tried to identify the temporal boundaries of the remaining 400 *unsegmented* remaining reports by comparing the text coming from corresponding archival descriptions (title + summary + keywords for instance), with the automatic speech transcription (ASR) of segments of the video file not already corresponding to one report (see Section 2.3). Simple similarity text measures such as the Jaccard distance, or *ratio* metrics in the Fuzzywuzzy Python package give encouraging but not entirely satisfying results. We plan to use a corpus-specific *TF-IDF* measure, or embedding methods such as word2vec or BERT in the future.

2.2 Typescripts: from page scans to structured textual data

Typescripts of the voice-overs have been linked to books and typescripts of each edition and separated with pages giving the summary of the edition (see Figure 2). This represents around 9,000 pages. At the beginning of the project, these documents were scanned in a good quality format (TIFF, color, 400 dpi). An optical character recognition (OCR) tool has thus been applied (Google Vision API in “Document” mode), giving spatially-located digital texts.

Once digitized, typescripts have to be separated from summaries. In order to achieve that, an automatic classifier has been trained by specializing the state-of-the-art Inception V3 classifier (Szegedy et al., 2016) with a few manually chosen examples. This gave about 2,600 pages of summaries and 6,400 pages of voice-overs.

LES ACTUALITÉS FRANÇAISES		
«REGARDS SUR LE MONDE»		
JOURNAL N° 17		
LE SPORT		
1. Le Relais à travers à Paris	31m	30
LA SEMAINE		
1. A Cannes, l'ouverture du 6ème Festival du film	16m	50
2. A Paris, Colette grand Officier de la Légion d'Honneur	13m	10
3. A Paris, une perspicacité qui paie <i>Concours d'ouvrages du "Piano"</i>	14m	20
JEU DE MAINS		
1. A Paris, l'élégance des mains	27m	60
DEMAIN		
1. Les "Castors de Thiais"	21m	10
2. Choisy le Roi aura sa cité moderne	21m	
3. A MontLouis, les fours solaires	29m	90
REGARDS SUR LE MONDE		
1. A Paris, la signature des accords	15m	50
2. Au Château de Windsor, l'anniversaire de la Reine	22m	50
3. Les fêtes de la création du soleil sur la colline de Sion (exclusif)	29m	70
GÉNÉRIQUE ET FIN		
	2m	20
DURÉE: 9' 8"	TOTAL:	251m 60
Sortie le 23 Avril 1953		
-:-:-:-:-		
LILLE		
1. La foire de Lille	32m	90
MAROC		
1. Tour du Maroc	93m	30
TUNISIE		
1. Elections caïdales	22m	60
ALGERIE		
1. Course des facteurs	26m	
BELGIQUE		
1. Bénédiction des vergers	14m	60
2. Radar	16m	50
3. Football Belgique-Hollande	38m	80
4. Tour du Maroc (V. courte)	40m	50
SARRE		
1. Coiffures	22m	50
2. Match de boxe	26m	
STRASBOURG		
1. Départ du paracout "Flandre"	12m	
2. Exposition du Vin		
3. Vignerons à l'Elysée	57m	
4. Le cinéma en relief (exclusif)	67m	
5. Record du monde sur moto	26m	70
REGIONAL L'UNION		
1. Les vignerons à l'Elysée	17m	

REGARDS SUR LE MONDE		Journal 17/53 2
1. A Paris, la signature des accords		53.166
A Paris, la signature des accords. Les pourparlers qui viennent de reprendre sont entrés enfin, dans la voie des réalisations. Le Général Lee Sang-cho qui conduit la délégation sino-coréenne, et l'Amiral Daniel, chef des délégués alliés, se sont rencontrés dans la "baraque de la paix" pour parapher l'accord sur l'échange des prisonniers malades et blessés. Et quelques minutes plus tard, l'Amiral Daniel pouvait agiter joyeusement le document signé, qui marque un premier pas vers la paix.		
2. A Windsor, l'anniversaire de la Reine		53.167
Au Château de Windsor, berceau des souverains britanniques, la Reine Elizabeth, refaisait un geste inauguré par Charles II, est venue à l'occasion de son anniversaire, remettre leur étendard aux grenadiers de la garde et recevoir leurs vœux dans les trois "heures" traditionnelles.		
3. Sur la colline de Sion, les fêtes de la création du soleil		53.168
Perpétuant une ancienne tradition, les fidèles ont attendu, sur les toits de la synagogue du Mont Sion, le lever du soleil. La croyance affirme en effet, que le soleil reprend tous les 28 ans, la place qu'il occupa au jour de sa création. Feront les musées, annoncé au son des trompes de David, l'astre, tel qu'il sortit des mains divines apparaît au peuple élu qui, manifeste sa joie en embrassant les rouleaux de la loi. De grandes réjouissances commencent alors, pour célébrer cet événement auquel peu de juifs peuvent se vanter d'avoir assisté trois fois dans leur vie.		
DEMAIN		
1. Les "Castors de Thiais"		53.163
Demain, trouverons nous à nous loger? Cette question si souvent répétée, les "Castors de Thiais" ont décidé de la résoudre par leurs propres moyens, en construisant eux-mêmes leur pavillon. Et, chaque soir, en sortant de leur travail, et durant leur congé hebdomadaire, des hommes venus de tous les métiers, se retrouvent sur leur chantier, aux portes de Paris, pour fabriquer des parpaings et participer à l'édification de la petite cité qu'ils habiteront demain. Lentement, mais sûrement, sous la conduite de l'ingénieur qui assure la direction du groupe, les murs s'élèvent et les maisons prennent forme. Pour les "Castors de Thiais", demain ne pose plus de problèmes de logement.		
2. Choisy le Roi aura sa cité moderne		53.164
Demain, également, Choisy le Roi sera dotée d'un magnifique ensemble de constructions modernes. Dernier vestige du passé, la cheminée d'une usine de faïencerie a été solennellement démolie. En présence d'un nombreux public, le Maire a mis le feu au brasier qui provoquait peu après, l'effondrement spectaculaire des 55 mètres de maçonnerie. Les démolisseurs ont terminé leur travail. Sur les 35.000 m ² qu'ils ont débarrassés, celui des bâtiments va commencer. Demain, une cité moderne blanche et aérée, qui comprendra 500 logements remplacera l'ancienne usine. Et, 500 familles auront enfin trouvé un foyer.		
3. Les fours solaires de Mont-Louis		53.165
Dans les Pyrénées, à 1600 mètres d'altitude, fonctionne à l'intérieur de la citadelle de Mont-Louis, le plus grand four solaire du monde. Cet appareil comprend un miroir orientable, composé de 500 glaces, commandé par une cellule photo-électrique, et un miroir parabolique		

Figure 2. Typescripts of voice-overs and summaries

Spatial and temporal alignment of transcripts. The objective of this alignment is to associate each report with the corresponding section of the typescripts. The available metadata allows processing this alignment year by year. This operation is done in two stages, by using on the one hand the result of the automatic speech transcription of the voice-over from the video files, and by using on the other hand the result of the OCR of the typescripts. The first step is done by minimizing a comparison measure between strings in order to find for each subject the corresponding typescripts page. The *partial ratio* method of the Fuzzywuzzy Python package allows looking for a partial inclusion of the speech-to-text into the OCR. Since topics and pages are approximately chronological, exhaustive searching is not required. The second step consists in spatially locating the text of the voice-over in the corresponding typescript page. For that, we use the alignment given by the Dynamic Time Warping algorithm (DTW), slightly modified to overcome the anchoring at the ends of the found path. The typescript area thus identified in the output of the OCR makes it possible to obtain the spatial coordinates of the commentary in the typewritten page. However, the method used does not allow locating transcripts overlapping over two pages. Additional treatment should be considered, for instance in order to get aligned text units for textometric analysis (see section 3.1).

2.3 Automatic audio analysis

The work on the audio part consists in detecting the speakers, transcribing speech into words (ASR) and detecting named entities (NE) using the systems we have developed for contemporary radio and television news.

Audio analysis of an old data set is an interesting challenge for automatic analysis systems. The recording devices used between 1945 and 1969 are very different from today's analog or digital devices. 35-mm films, which contain both sound and image, deteriorated before being digitized in the

2000s. Moreover, the acoustic and language models are generally trained on data produced between 1998 and 2012. This 50-year time gap has consequences on the system's performance.

Technically, acoustic models for ASR and speakers were trained on about 300 hours drawn from several sources of French TV and radiophonic broadcast news² with manual transcripts. The ASR language models were trained on these manual transcripts, French newspapers, news websites, Google news and the French GigaWord corpus, for a total of 1.6 billion words. The vocabulary of the language model contains the 160k most frequent words. The NE models were trained only on a subset of manual transcripts³.

Prior to the transcription process, the signal is cut into homogeneous speech segments and grouped by speakers. We refer to this process as the Speaker Diarization task. Speaker Diarization is first applied at the edition level, where each video record is separately processed. Then, the process is applied at the collection level, over all the 1,200 editions, in order to link the recurrent speakers. The system is based on the LIUM S4D toolkit (Broux et al., 2018), which has been developed to provide homogeneous speech segments and accurate segment boundaries. Purity and coverage of the speaker clusters are also one of the main objectives. The system is composed of acoustic metric-based segmentation and clustering followed by an i-vector-based clustering applied to both edition and collection levels.

The ASR system is developed using the Kaldi Speech Recognition Toolkit (Povey et al., 2011). Acoustic models are trained using a Deep Neural Network which can effectively deal with long temporal contexts with training times comparable to standard feed-forward DNNs (chain-TDNN (Povey et al., 2016)). Generic 3 and 4-gram language models, which allow users to compute the probability of emitting one word knowing a history of 2 or 3 words, were also trained and used during decoding. To help the reading, two sequence labeling systems (Conditional Random Field models) have been trained over manual transcripts to add punctuation and upper-case letters respectively.

The NE system, based on the NeuroNLP⁴ toolkit, helps the text analysis. The manually annotated transcripts are used to train a text-to-text sequence labeling system. The system detects eight main entity types: amount, event, function, location, organization, person, product and time.

ASR was performed on the full collection of 1,200 national editions in order to feed Okapi and TXM platforms for historians' analyses (see Section 3): about 300 hours of video, resulting in more than 1.5 million words. A subset of 12 editions from 1945 to 1969 were manually transcribed to evaluate the audio analysis systems. Due to the 50-year time gap, human annotators had some difficulties with the spelling of NE, especially regarding people and foreign NE. Thanks to Wikipedia and INA thesaurus, most of NEs have been checked. However, speakers are very hard to identify. Most of them are male voice-overs. Their faces are never seen and their names are rarely spoken, nor displayed on the images. Only journalists performing interviews and well-known people, such as politicians, athletes and celebrities, can be accurately identified and named.

The quality of an ASR system is evaluated using the so-called Word Error Rate (WER). This metric consists of counting the number of insertions, deletions and substitutions of words between the transcripts automatically generated by the ASR and the human transcripts considered as an oracle. The WER is 24.27% on ANTRACT data using the generic ASR system trained on modern data. The same system evaluated on 2010 data⁵ achieves 13.46%. It is known that ASR systems are sensitive to acoustic and language variations between train corpus and test corpus. Here, the WER is almost double. It is generally difficult to exploit transcripts in a robust way when WER is above 30%. Most of the errors come from unknown words (which are not listed in the 160k vocabulary). These out of vocabulary words (OOV) are confused with acoustically close words, which have a negative impact on neighboring words. The system always selects the most likely word sequence containing the word replacing the OOV.

Additional contemporary data, such as archival descriptions and typescripts, would be useful to adapt the language model. Therefore, abstracts, titles and descriptions have been extracted from the archival descriptions. OCR sentences (see Section 2.2) have been kept when at least 95% of the words belong to the ASR vocabulary. A "in domain" training corpus composed of 1.3 million words from archival descriptions and 4.7 million words from typescripts was built. The 4,000 most frequent words were selected to train the new ANTRACT language model, which reduces the error rate by half: from 24.27% to 12.06% WER. Figure 3 shows a sample of automatic transcription of the July 14, 1955 edition. The gain is significant thanks to the typescripts which are very similar to manual transcriptions. This "in domain" training corpus is contrary to the rules usually set during the well-known ASR system evaluations: a test data set should never be used to build a training corpus. However, in our case, the main goal is to provide the best transcripts to historians.

Future work will focus on ASR acoustic models improvement. We plan to use an alignment of typescripts with the editions, as well as historian users' feedback providing some manually revised transcriptions. The objective is to select zones of confidence to be added to the learning data. Evaluation of the Named Entities is the next step in the roadmap. The speaker evaluation will be more difficult because of their identities, which are not available. We plan to evaluate both the detection of voice-overs and interviewers. Furthermore, some famous persons, selected in collaboration with historians for their relevance in historical analyses, will also be identified, with the possible help of crossing results with image analysis as described in Section 2.4.

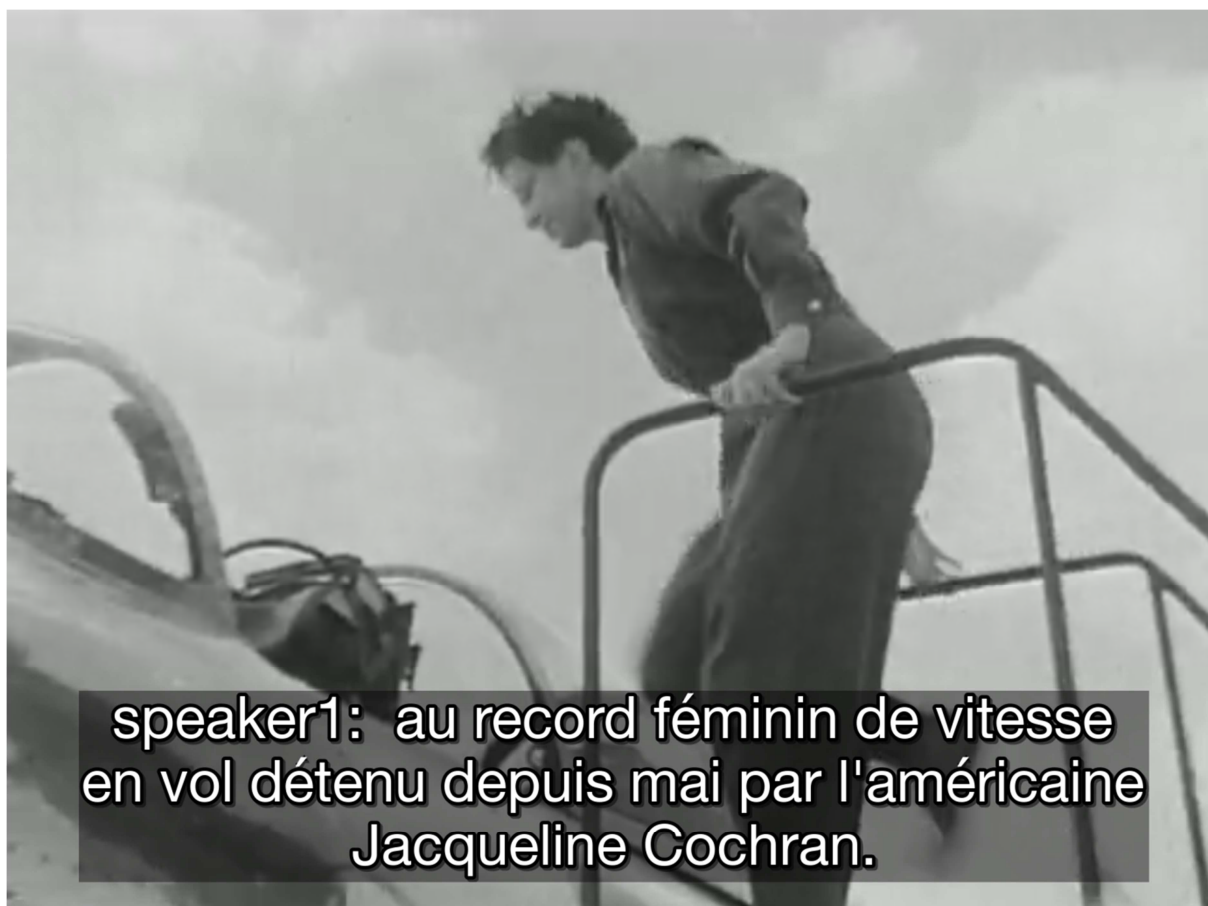


Figure 3. Sample “Actualité Francaise July 14, 1955 from 6:06 to 6:49”. Subtitle is an ASR file with in domain language model, automatic punctuation and upper case. @INA

2.4 Automatic visual analysis

Identifying the people appearing in a video is undoubtedly an important cue for its understanding. Knowing who appears in a video, when and where, can also lead to learning interesting patterns of relationships among characters for historical research. Such person-related annotations could provide ground for value added content. An historical archive such as the *Actualités Françaises* corpus contains numerous examples of celebrities appearing in the same news segment as De Gaulle and Adenauer (see Figure 4). However, the annotations produced manually by archivists do not always identify with precision those individuals in the videos. On the other side, the web offers an important amount of pictures of those persons, easily accessible through Search Engines using their full name as search terms. In ANTRACT, we aim to leverage these pictures for identifying faces of celebrities in video archives.



Figure 4. De Gaulle and Adenauer together in a video from 1959. @INA

There has been much progress in the last decade regarding the process of automatic recognition of people. It generally includes two steps: first, the faces need to be detected (i.e. which region of the frame may contain a person face) and then recognised (i.e. to which person this face belongs to).

The Viola-Jones algorithm (Viola, 2004) for face detection and Local Binary Pattern (LBP) features (Ahonen, 2006) for the clustering and recognition of faces were the most famous techniques used until the advent of deep learning and convolutional neural networks (CNN). Nowadays, two main approaches are in use to detect faces in video and both are using CNNs. The Dlib library (King, 2009) provides good performance for frontal images but it requires an additional alignment step (which can also be performed using the Dlib library) before face recognition can be performed. The recent Multi-task Cascaded Convolutional Networks (MTCNN) approach provides even better performance using an image-pyramid approach and integrates the detection of face landmarks in order to re-align detected faces to the frontal position (Zhang, 2016).

Having located the position and orientation of the faces in the video images, the recognition process can be performed in good conditions. Several strategies have been detailed in the literature to achieve recognition. Currently, the most practical approach is to perform face comparison using a transformation space in which similar faces are close together, and to use this representation to identify the right person. Such embeddings, computed on a large collection of faces, are often available to the research community (Schroff, 2015).

Within ANTRACT, we developed an open source Face Celebrity Recognition system. This application is made of the following modules:

- A web crawler which, given a person's name, automatically downloads from Google a set of k photos that will be used for training a particular face model. In our experiments, we generally use $k = 50$. Among the results, the images not containing any face or containing more than one face are discarded. In addition, end users (e.g. domain experts) can manually exclude wrong results, for example, corresponding to pictures that do not represent the searched person.
- A training module where the retrieved photos can be converted to black-and-white, cropped and resized in order to obtain images only containing a face, using the MTCNN algorithm (Zhang, 2016). A pre-trained Facenet (Schroff, 2015) model with Inception ResNet v1

architecture trained on VGGFace2dataset (Cao, 2018) is applied in order to extract visual features of the faces. The embeddings are used to train a SVM classifier.

- A recognition module where a newsreel video is received as input and from which all frames are extracted at a given skipping distance d (in our experiments, we generally set $d = 25$, namely 1 sample frame per second). For each frame, the faces are detected (using the MTCNN algorithm) and the embeddings computed (Facenet). The SVM classifier decides if the face matches the ones among the training images.
- Simple Online and Realtime Tracking (SORT) is an object tracking algorithm, which can track multiple objects in real-time (Bewley, 2016). Its implementation is inspired by the suggestion code from Linzaer⁶. The algorithm uses the MTCNN bounding box detection and tracks it across frames. We introduced this module to increase the robustness of the library. By introducing this module, while making the assumption that faces do not swap coordinates across consecutive frames, we aim to get a more consistent prediction.
- Finally, the last module groups together the results coming from the classifier and the tracking modules. We observe that even though the face to recognize remains the same over consecutive frames, the face prediction sometimes changes. For this reason, we select for each tracking the most frequently occurring prediction, taking also into account the confidence score given by the classifier. In this way, the system provides a common prediction for all the frames involved in a tracking, together with an aggregated confidence score. A threshold t can be applied to this score in order to discard the low-confidence prediction. According to our experiments, $t = 0.6$ gives a good balance between precision and recall.

In order to make the software available as a service, we wrapped it into a RESTful web API, available at <http://facerec.eurecom.fr/>. The service receives as input the URI of a video resource, as it appears in Okapi, from which it retrieves the media object encoded in MPEG-4. Two output formats are supported: a custom JSON format and a serialization format in RDF using the Turtle syntax and the Media Fragment URI syntax (Troncy et al., 2012), with normal play time (npt) expressed in seconds to identify temporal fragments and $xywh$ coordinates to identify the bounding box rectangle encompassing the face in the frame. A third format, again following the Turtle syntax, will be soon implemented so that the results can be directly integrated in the Okapi Knowledge Graph. A light cache system is also provided in order to enable serving pre-computed results, unless the no cache parameter is set which is triggering a new analysis process.

We run experiments using the face model of Dwight D. Eisenhower on a selection of video segments extracted from Okapi, among the ones that have been annotated with the presence of the American president using the *ina:imageContient* and *ina:aPourParticipant* properties in the knowledge graph. In the absence of a ground truth, we performed a qualitative analysis of our system on three videos. For each detected person, we manually assessed whether the correct person was found or not. Out of the 90 selected segments, the system correctly identified Eisenhower in 33 of them. However, we are not sure that Eisenhower is effectively visually present in all 90 segments. We are currently working on extracting from the ANTRACT corpus a set of annotated segments to be used as ground truth so that it is possible to measure the precision and recall of the system.

In addition, we made the following observations:

- The library generally fails in detecting people when they are in the background, or when the face is occluded.
- When faces are perfectly aligned, they are easier to detect. Improvements on the alignment algorithm are foreseen as future work.

- When setting a high confidence threshold, we do not encounter cases where we confuse one celebrity by another one. Most errors are about confusing an unknown face with a celebrity in the dataset.

In order to easily visualize the results and to facilitate history scholars' feedback, we developed a web application that shows the results directly on the video, leveraging on HTML5 features. The application also provides a summary of the different predictions, enabling the user to directly jump to the relative part of the video where the celebrity appears. A slider allows changing the confidence threshold value, in order to better investigate the low-confidence results.

The application is publicly available at <http://facerec.eurecom.fr/visualizer/?project=antract> (see Figure 5).

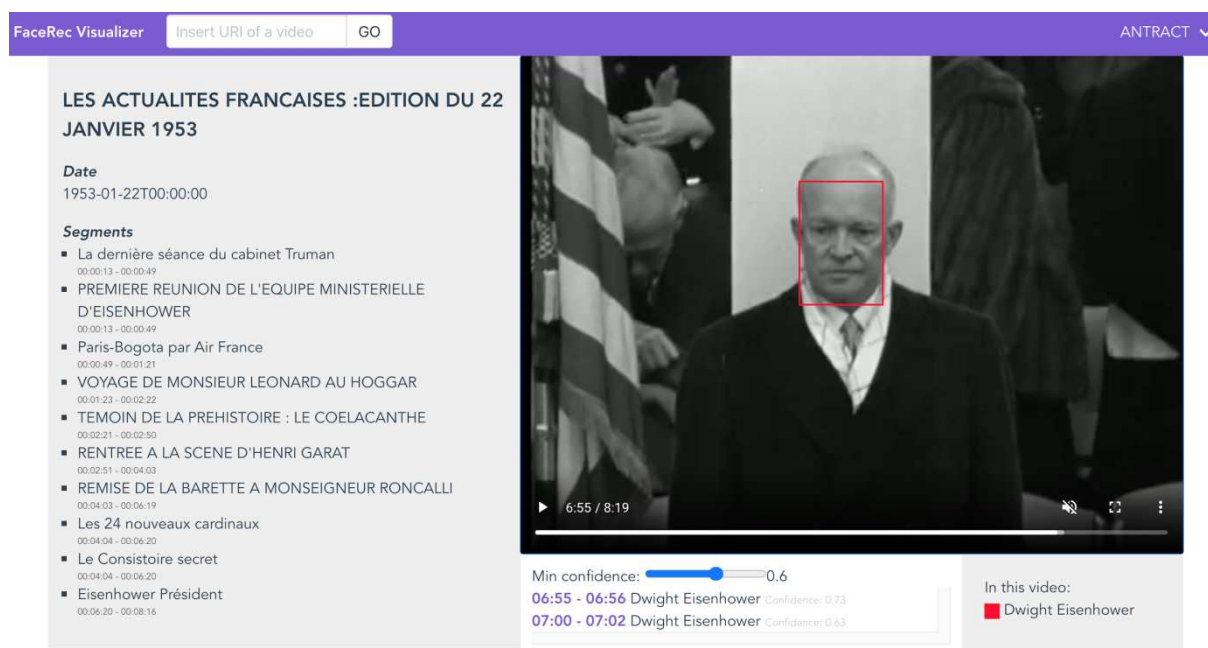


Figure 5: The visualizer of the Celebrity Face Recognition System

3. Platforms for historians' exploration and analysis of the corpus

The corpus built with automatic tools in section 2 is explored interactively by historians using two platforms:

- the TXM platform for analysis of text corpora based on quantitative and qualitative exploration tools, and augmented during the ANTRACT project to facilitate the link between textual data and audio and video data;
- the Okapi knowledge-driven platform for the management and annotation of video corpora using semantic technologies.

3.1 The TXM platform for interactive textometric analysis

Text analysis is achieved through a textometric approach (Lebart et al., 1998). Textometry combines both quantitative statistical tools and qualitative text searching, reading and annotating. On the one hand, statistical functionalities include keyword analysis, collocations, clustering and correspondence analysis. This makes a significant analytical power addition in comparison with usual annotation and search & count features in audiovisual transcription software such as CLAN (MacWhinney, 2000) or

ELAN (ELAN, 2018). On the other hand, yet again in the textometric approach, qualitative analysis is carried out by advanced KWIC concordancing, by placing an emphasis on easy-access to high quality of layout rendering of source documents and by providing annotation tools. Such a qualitative side is marginal if not absent in conventional text mining applications (Hotho et al., 2005; Feinerer et al., 2008; Weiss et al., 2015): most of them process plain text, getting rid of text body markup, if any, and aim at synthetic visualization displacing close text reading.

Textometry is implemented by the TXM software platform (Heiden, 2010). TXM is produced as an open-source software, which integrates several specialized components: R (R Core Team, 2014) for statistical modeling, CQP for full text search engine (Christ, 1994), TreeTagger (Schmid, 1994) for Natural Language Processing (morphosyntactic tagging and lemmatization). TXM is committed to data and software standardization and sharing efforts, and has notably been designed to manage richly-encoded corpora, such as XML data and TEI⁷ encoded texts ; for ANTRACT textual data, TXM imports tabulated data (Excel format export of tables from INA documentary databases) and files in the Transcriber XML format provided by speech-to-text software (see Section 2.3). TXM is dedicated to text analysis, but also helps to manage multimedia representations associated with the texts, whether it is scanned images of source material, audio or video recordings: actually, these representations participate in the interpretation of TXM common tools results in their full semiotic context.

In 2018, we began to build the AFNOTICES TXM corpus by importing the INA archival descriptions: each news report is represented by several textual fields (title, abstract, sequence description) and several lexical fields (keyword lists of different types such as topics, people, or places, and credits with names of people shown or cameramen) and labeled by a dozen metadata (identifier, broadcast date, film producer, film genre, etc.) which are useful to contextualize or categorize reports.

In 2019, we began the production of the AFVOIXOFFV02 TXM corpus which makes the voice-over transcripts (see Section 2.3) searchable and available for statistical analysis, synchronized at the word level for video playback and labeled by INA documentary fields.

These corpora may still be augmented by aligning new textual modalities: texts from narration typescripts (OCR text and corresponding regions in the page images) (see Section 2.2), annotations on videos (manual annotations added by historians through the Okapi platform (see Section 3.2), as well as automatic annotations generated by image recognition software (see Section 2.4), named entities, etc.

One of the technical innovations achieved for the project has been the consolidation of TXM back-to-media component (Pincemin et al., 2020), so that any word or text passage found in the result of a textometric tool can be played with its original video; we have also implemented authenticated streamed access to video content from the Okapi media server, which happened to be a key development for video access given the total physical size and the security constraints of such film archive data.

The following screenshots illustrate typical textometric analysis moments of current studies within the ANTRACT project.

In Figure 6 and Figure 7, we study the context of use for the word “*foule*” (crowd), through a KWIC concordance. A double-click on a concordance line opens up a new window (on the right-hand side) which displays the complete transcript in which the word occurs. Then, we click on the music note symbol at the beginning of the paragraph to play the corresponding video. A dialog box prompts for credentials before accessing the video on the Okapi online server. This opportunity to confront textual analysis with the audiovisual source is all the more important here because textual data were generated by the speech-to-text automatic component, whose output could not be fully revised. Moreover, the video may add significant context that is not rendered in plain text transcription.

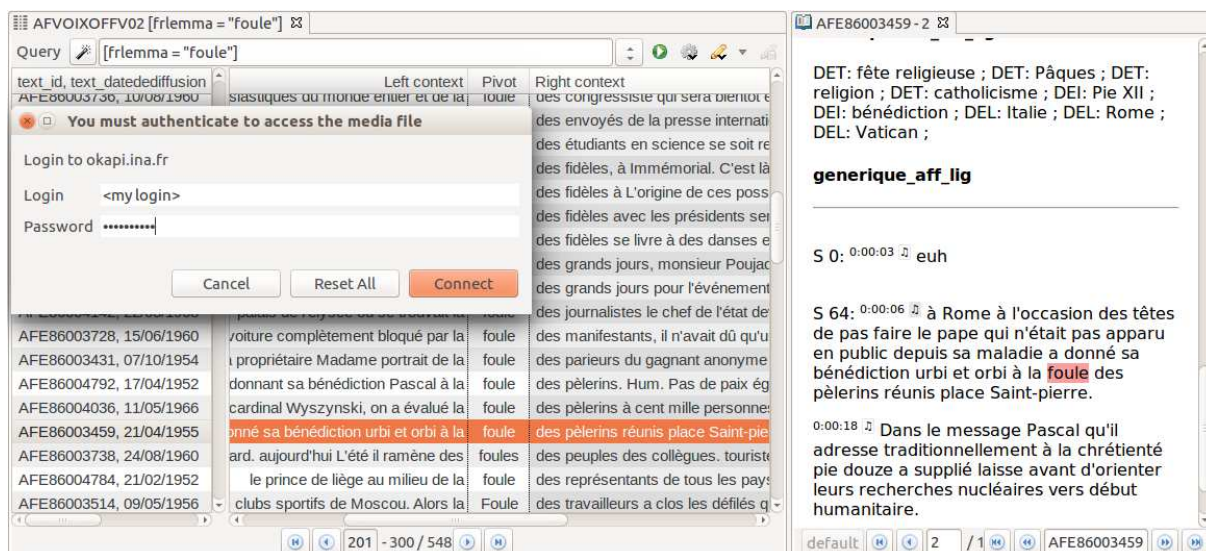


Figure 6. CONCORDANCE of the word “*foule*” (crowd) in the voice-over corpus (left window), voice-over transcript EDITION corresponding to the selected concordance line (right window), and the authentication dialog box to access the Okapi video server to play the video at 0:00:06 (top left window).



Figure 7. Hyperlinked windows managing results associated with the word “*foule*” (crowd): CONCORDANCE (left window), transcript EDITION (middle window) and synchronized video playback (right window)

Our second example is about the place of agriculture and farmers in the *Actualités françaises*, and how the topic is presented. It shows how one can investigate if a given word has the same meaning in documentation and in commentary, or if different words are used when dealing with the same subject. We first get (Figure 8) a comparative overview of the quantitative evolution of occurrences from two word families, derived from the stems of “*paysan*” and “*agricole*”/ “*agriculture*” (see detailed list of words in Figure 9, left hand side window). We complete the analysis with contextual analysis through KWIC concordance views (see Figure 8, lower window) and cooccurrences computing (see Figure 9). We notice that “*paysan*” becomes less used from 1952 onwards, and that it is preferred to “*agriculteur*” when speaking of the individuals present in the newsreels extracts; conversely, “*agricole*”/ “*agriculture*” are used in a more abstract way, to deal with new farm equipment and socio-economic transformation of this line of business.

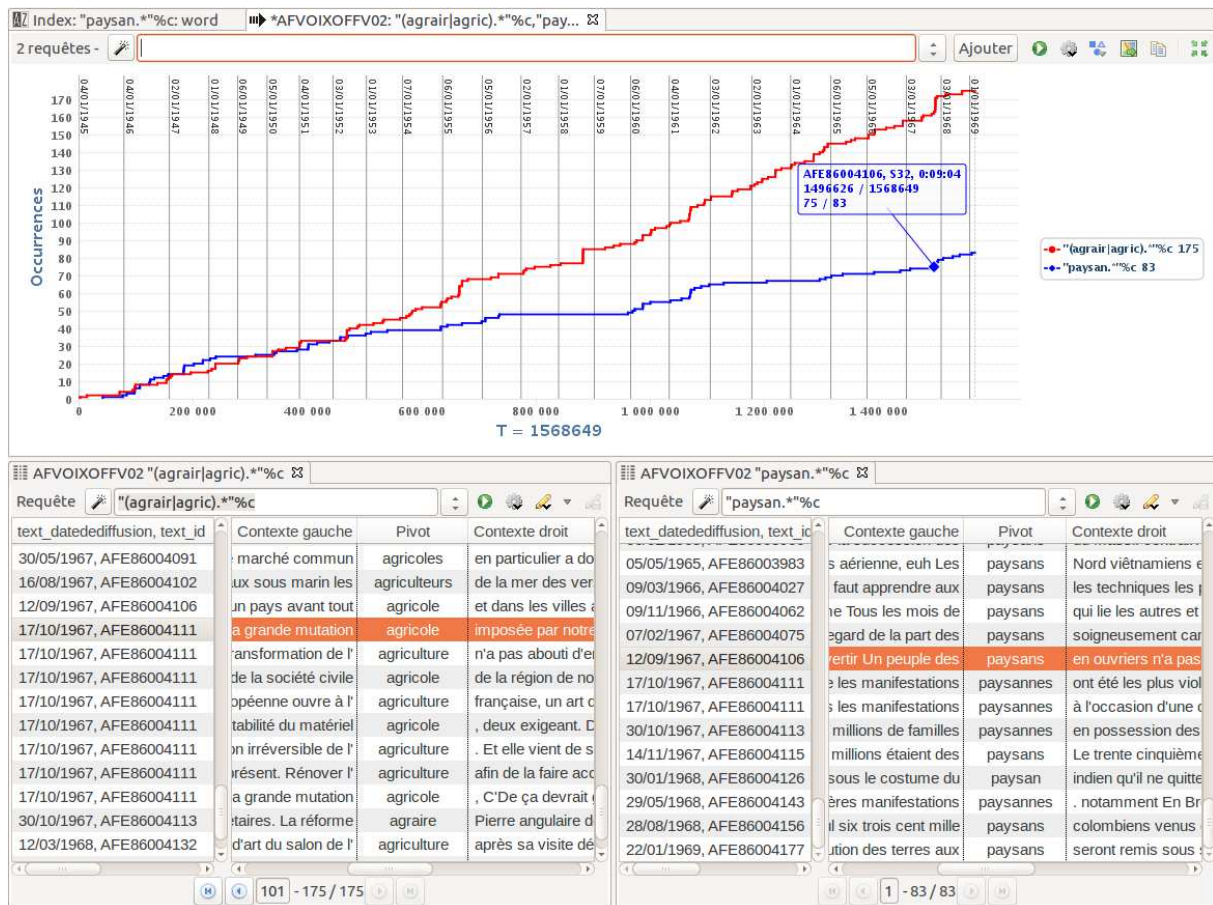


Figure 8. PROGRESSION chart (upper window), and hyperlinked KWIC CONCORDANCES (lower window), to compare two word families related to farming

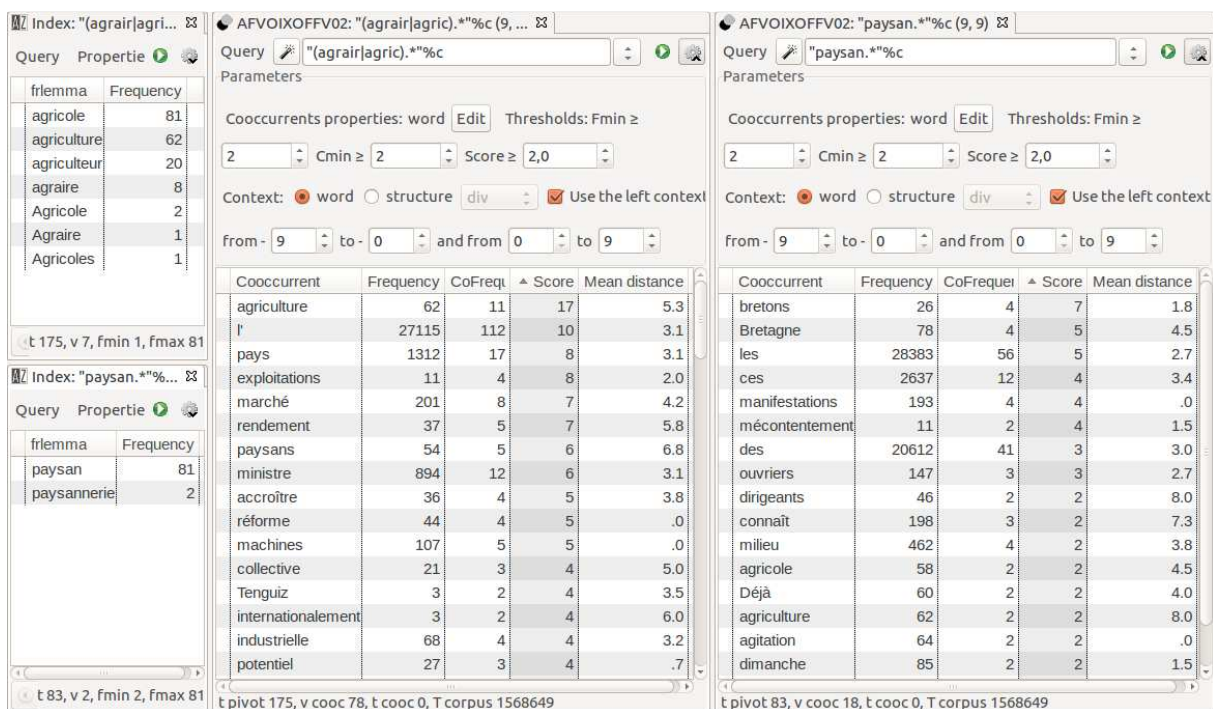


Figure 9. INDEX results detailing the content of two word families (left margin), and COOCCURRENCES statistical analysis to characterize their contexts

Combining word lists (INDEX) and morphosyntactic information is very effective to summarize phrasal contexts. For instance, in Figure 8, we can compare which adjectives qualify “*foule*” in the archival descriptions, and which ones qualify “*foule*” in the voice-over speeches. For a given phrase (“*foule immense*”, huge crowd) in the voice-over, we compute its cooccurrences in order to identify in which kind of circumstances the phrase is preferred (funerals, religious meetings). In TXM, full-text search is powered by the extensive CQP search engine (Christ, 1994), which allows very fine-tuned and contextualized queries.

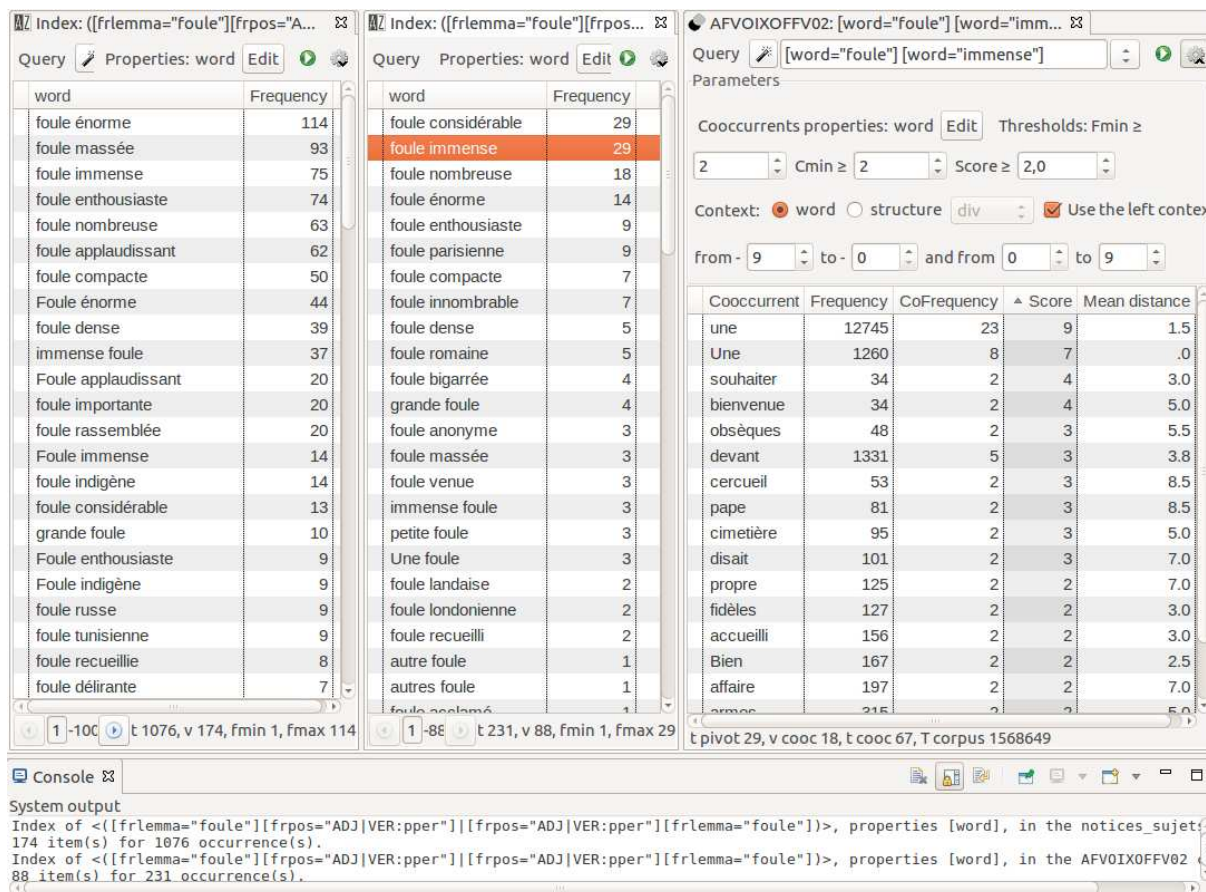


Figure 10. INDEX of “*foule*” (crowd) preceded or followed by an adjective, in archival descriptions (left window) or in voice-over transcripts (middle window). COOCCURRENCES for “*foule immense*” (huge crowd) in voice-over transcripts (right window)

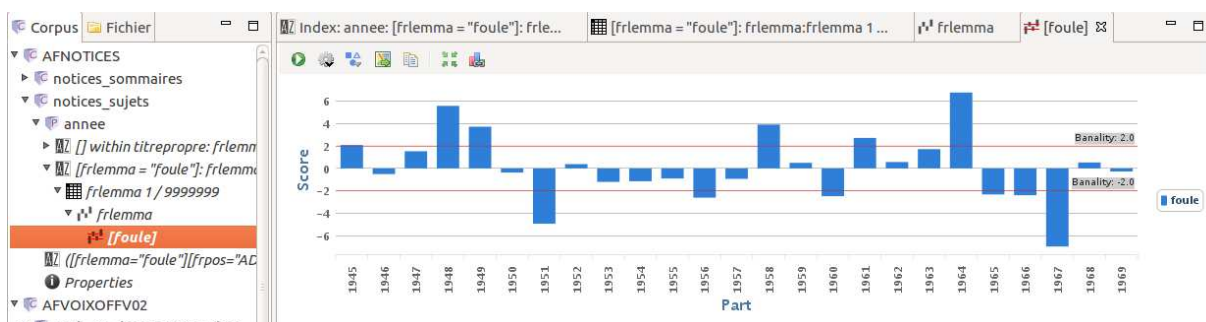


Figure 11. Statistical SPECIFICITY chart for “*foule*” (crowd) over the years

For chronological investigations, we can divide the corpus into time periods in a very flexible way, such as years or groups of years. Any encoded information may be used to build corpus subdivisions. Then the SPECIFICITY command — that implements a Fisher’s Exact Test, known as one of the best

calculations to find keywords (McEnery and Hardie, 2012)— statistically measures the steady use, or the singular overuse or underuse of any word. The function can also be used to bring to light the specific terms for a given period, or for any given part of the corpus. For example, (Figure 9) focuses on the word “foule” over the years. Peak years reveal important political events (e.g. the liberation of France after WW2, the advent of the Fifth Republic), which match the high exposure of Général de Gaulle. However, the most frequent occurrences do not necessarily correspond to political upheavals.

The figure consists of two screenshots of a software interface, likely a statistical analysis tool, showing resonance analysis results. Both windows have a 'Property' dropdown set to 'word' and a search icon.

Upper Window: Resonance analysis for 'foule_in_documentary_desc t=396023'

Units	Frequency T 1568649	foule_in_documentary_desc t=396023	index
foule	515	353	93.6
président	1865	830	72.0
Gaulle	708	375	55.1
général	1750	731	50.8
république	782	344	29.4
la	44672	12268	26.9
accueil	194	119	25.5
cortège	150	99	25.0
enthousiasme	151	96	22.4
avait	2528	860	22.3
devant	1331	489	19.9
était	3789	1208	19.6
peuple	469	208	18.6
acclamations	67	52	18.4

Lower Window: Resonance analysis for 'foule_in_documentary_desc n voice_without_gaulle_president t=264308'

Units	Frequency T 1568649	foule_in_documentary_desc n voice_without_gaulle_president t=264308	index
foule	515	211	37.5
peloton	215	100	23.2
départ	719	223	20.1
minutes	612	182	14.6
étape	323	113	14.6
princesse	206	82	14.3
course	381	125	13.6
coureurs	129	58	13.0
roi	448	138	12.6
devant	1331	328	12.5
personnes	333	109	11.9
reine	354	114	11.9
carnaval	48	30	11.7
corrida	43	28	11.6

Figure 12. Example of resonance analysis (Salem, 2004): SPECIFIC terms in voice-over comments for reports showing a crowd (according to archival description) (upper window) ; then, SPECIFIC terms in voice-over comments for reports showing a crowd and having no mention of De Gaulle or “président” (president) (lower window)

With Figure 12, we apply a statistical resonance analysis (Salem, 2004). When a crowd is shown (as indicated by the archival description), what are the most characteristic words said by the voice-over? “Président” and “[le général De] Gaulle” represent the main context (Figure 12, upper window). In a second step, we remove all the reports containing one of these two words and focus on the remaining reports to bring out new kinds of contexts associated with the view of a crowd (Figure 12, lower window), such as sports, commemorative events, demonstrations, festive events, etc. The recurring

term “*foule*” (crowd) in the voice-overs promotes a sense of belonging to a community of fate. From a methodological perspective, this kind of cross-querying combined with statistical comparison between textual newsreel archival descriptions and commentary transcripts helps investigate correlations or discrepancies between what is shown in the newsreels and what is said in their commentaries. Such a combination of statistics across media is rarely provided by applications.

Figure 13 provides a first insight of a correspondence analysis output: we computed a 2D-map of the names of people who are present in more than 20 reports, in relation with the years in which they are mentioned. We thus get a synthetic view of the relationship between people and time in the *Actualités françaises* reports. In terms of calculation, as textometry often deals with frequency tables crossing words and corpus parts (here we crossed people’s names and year divisions), it then opts for correspondence analysis, because this type of multidimensional analysis is best suited to such contingency tables (Lebart et al., 1998).

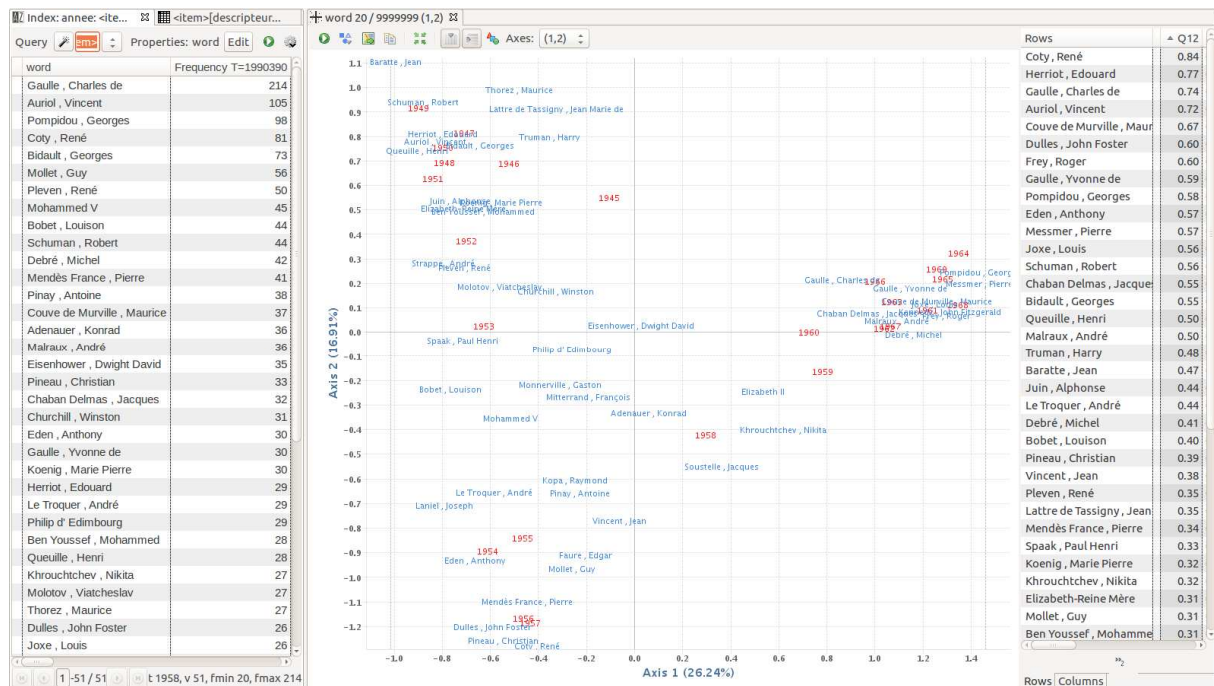


Figure 13. CORRESPONDENCE ANALYSIS (first plane) of the frequency table crossing the years and the names of 51 people that are present in at least 20 reports

3.2 Okapi platform for interactive semantic analysis

Okapi (Open Knowledge Annotation and Publication Interface) (Beloued et al., 2017) is a knowledge-based online platform for semantic management of content. It is at the intersection of three scientific domains: Indexing and description of multimedia content, knowledge management systems and Web content management systems. It takes full advantage of semantic web languages and standards (RDF, RDFS, OWL (Motik et al., 2012)) to represent content as graphs of knowledge; it applies semantic inferences on these graphs and transforms them to generate new hypermedia content like web portals.

Okapi provides a set of tools for analyzing multimedia content (video, image, sound) and managing corpora of annotated video and sound excerpts as well as image sections. Analysis tools allow the semantic indexing and description of content using domain ontology. The corpus management tools provide services for the constitution and visualization of thematic corpora as well as their annotation and enrichment in order to generate mini-portals or thematic publications of their contents.

The Okapi's knowledge management system stores knowledge as graphs of named entities and provides services to retrieve, share and present them as linked open data. These entities can be aligned with other entities in existing knowledge bases like dbpedia and wikidata and so makes Okapi interoperable with the Linked Open Data ecosystem (Bizer et al., 2009). The named entities can be of different types and categories and vary according to the studied domain. For instance, for audiovisual archives, entities may concern persons, geographical places and concepts.

Finally, the Okapi's Content Management System (Okapi's CMS) considers the characteristics of the studied domain and user preferences to generate web interfaces and tools for Okapi as well as content portals adapted to the domain. This publishing framework allows also authors to focus on their authoring work and to create thematic portals without any technical skills. The author can specify his thematic publication as a set of interconnected multimedia elements (video, image, sound, editorial texts). The framework applies thereafter a set of publishing rules on these elements and generates a web site.

The Okapi platform is used by historians to constitute thematic corpora and to publish their portals as explained in the above paragraphs. Okapi can also be used by researchers in computer science and data scientists to show and improve the results of their automatic algorithms (face detection and recognition, automatic speech recognition, etc.). The following sections show some examples of how the Okapi platform can be used on the collection “*Les Actualités Françaises*” (AF) in the context of the ANTRACT project.

The media analysis can be carried out manually by annotators or automatically by algorithms on several axes as shown in Figure 14. In this example, thematic analysis (the layer entitled “*strate sujets*”) of the AF program “*Journal Les Actualités Françaises : émission du 10 juillet 1968*” consists in identifying the topics addressed in this program, their temporal scope and a detailed description of the topic in terms of the subject we are talking about, the places where it happens and the persons who are involved in this subject.

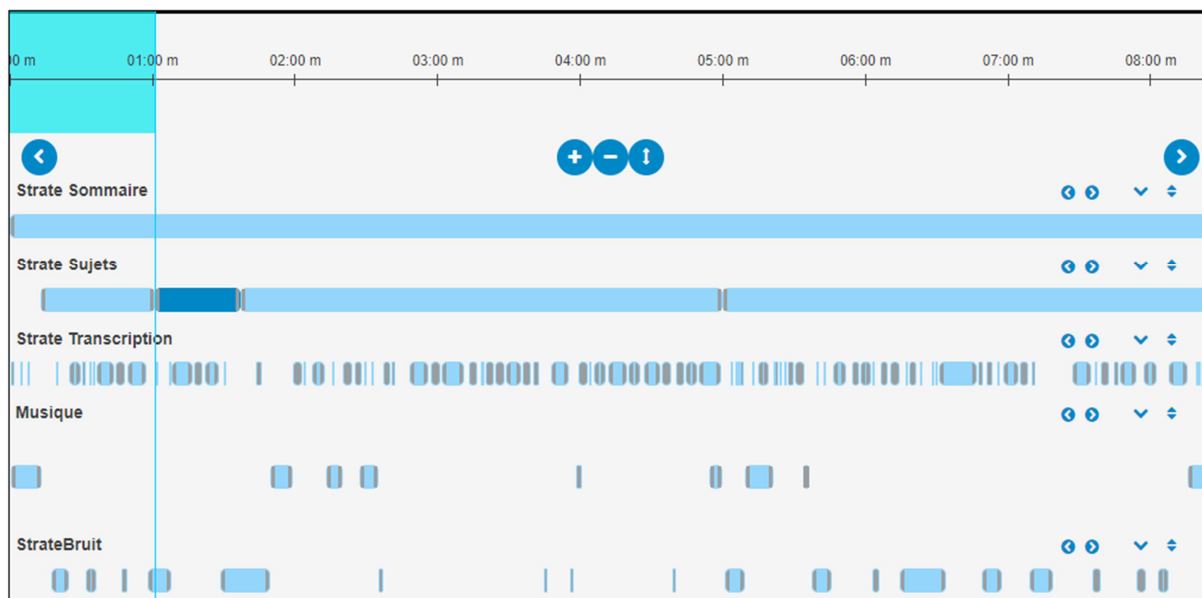


Figure 14. Timeline for Media Analysis

The user can create and remove analysis layers and their segments as well as the description of each segment and its timecodes. Considering the second segment in the example where we are talking about the **water sports (concept)** in **England (Place)**, especially the adventures of the solo sailor **Alec Rose (Person)** as indicated in the following form (Figure 15): The user can edit this form to change and

create new description values of the selected segment. These concepts, places and persons are a subset of named entities that are managed and suggested by Okapi to complete the description of the segment.

The screenshot shows the 'Segment Metadata Form' with the following sections and content:

- résumé**: A list of video descriptions including 'VG du voilier du navigateur solitaire, Alec ROSE, naviguant dans le port de Portsmouth, escorté par d'autres bateaux', 'VG PANO en plongée sur une foule dense de spectateurs massés sur les quais du port, certains agitant des drapeaux anglais', 'VG avec ZAV sur le voilier "LIVELY LADY" : Alec ROSE sur le pont, prêt à lancer les amarres', 'VG de la foule venue l'accueillir', 'PM du navigateur solitaire Alec ROSE, en costume et casquette de marin, embrassant sa femme sur le quai de Portsmouth et faisant un geste de salut', 'VG de nombreux spectateurs agitant la main', 'VG du maire de Portsmouth, près de Alec ROSE et de sa femme, donnant le signal des "Hurrah !"', and 'foule acclamant'.
- thème**: A search bar, a language dropdown set to 'Français', and a list item 'sport nautique(Schéma Noms communs)' with a plus icon and a close icon.
- à l'image**: A search bar, a language dropdown set to 'Français', and a list item 'Rose, Alec' with a plus icon and a close icon.
- lieu**: A search bar, a language dropdown set to 'Français', and a list of locations: 'Grande Bretagne', 'Royaume Uni', and 'Angleterre', each with a plus icon and a close icon.

Figure 15. Segment Metadata Form

The other analysis layers (transcription, music detection, etc.) are provided by automatic algorithms. The metadata provided by these algorithms can enrich the ones created manually by users and can be used by the Okapi platform to generate a rich portal that brings value to the content and provides several access and navigation possibilities in the content as shown in Figure 16.

Journal Les Actualités Françaises : émission du 10 juillet 1968

ANALYSE

00:01:04.06
00:08:26.04

STRATE SOMMAIRE

STRATE SUJETS

MUSIQUE

VOIX HOMME

Rechercher

SOMMAIRE

GÉNÉRALITÉS

DESCRIPTEURS

résumé

- VG du voilier du navigateur solitaire, Alec ROSE, naviguant bateaux - VG PANO en plongée sur une foule dense de spectat des drapeaux anglais - VG avec ZAV sur le voilier "LIVELY LADY" VG de la foule venue l'accueillir - PM du navigateur solitaire embrassant sa femme sur le quai de Portsmouth et faisant agitant la main - VG du maire de Portsmouth, près de Alec ROSI foule acclamant.

thème

▶ sport nautique

à l'image

▶ Rose, Alec

lieu

▶ Angleterre

▶ Grande Bretagne

▶ Royaume Uni

Au Stade Charlety, la confrontation des athlètes Américains et Français

durée: 00:00:48:0

Alec Rose, après 354 jours sur un bateau : "la terre est ronde"

durée: 00:00:36:0

Figure 16. Okapi portal page of the AF news “Journal Les Actualités Françaises: émission du 10 Juillet 1968”.

The generated metadata are also used as advanced criteria for looking for video excerpts and so allow users to constitute their thematic corpora focused on some topics. Figure 15 shows an example of an advanced search of segments which talk about “**Water sports**” in “**England**”. Like all Okapi’s objects, a query is represented as a knowledge graph and then transformed into a SPARQL query. The results of this query, illustrated by Figure 17 and 18, can be used to create a corpus.

The interface is titled 'Notice sujet' with a search bar labeled 'Rechercher'. Below the title is a navigation bar with buttons: 'Créer', 'Gérer', 'Rechercher', 'Valider', '</>', 'Quitter', and two icons. The 'DESCRIPTEURS' tab is selected. The form contains two sections: 'thème' and 'lieu'. The 'thème' section has a dropdown menu showing 'sport nautique (Schéma Noms communs)(Schéma Noms communs)' and a search bar with a 'Rechercher' icon and a close 'x' icon. The 'lieu' section has a dropdown menu showing 'Angleterre' and a search bar with a 'Rechercher' icon and a close 'x' icon. Both search bars have a language dropdown set to 'Français'.

Figure 17. Example of an Okapi Query

The interface is titled 'Notice sujet (3)'. It displays a list of three query results, each with a dropdown arrow, a checkbox, and a title. The first result is 'ROBERT MANRY, 48 ANS : TRAVERSEE SOLITAIRE DE L'OCEAN', the second is 'La course de grands voiliers Torbay- Rotterdam', and the third is 'Alec Rose, après 354 jours sur un bateau : "la terre est ronde"'. Each result has a 'Rechercher' icon and a 'Gérer' icon.

Dropdown	Checkbox	Title	Rechercher	Gérer
⌵	<input checked="" type="checkbox"/>	ROBERT MANRY, 48 ANS : TRAVERSEE SOLITAIRE DE L'OCEAN		
⌵	<input checked="" type="checkbox"/>	La course de grands voiliers Torbay- Rotterdam		
⌵	<input checked="" type="checkbox"/>	Alec Rose, après 354 jours sur un bateau : "la terre est ronde"		

Figure 18. Example of query results

The corpus itself is an object to be annotated, i.e, the user can add new metadata on the corpus itself or on its elements (video excerpts) and put rhetorical relations between them. Figure 19 shows a corpus of three excerpts, retrieved from the query presented in the previous paragraph. It displays also a rhetorical relationship between the two segments: “*Robert Manry, 48 ans: Traversée solitaire de l’océan*” which illustrates the other segment “*Alec Rose, après 354 jours sur un bateau: “la terre est ronde”*”. All these metadata will be used to create a thematic portal focused on the content of the corpus or integrated into a story through the inclusion of editorial content and preferred reading paths.

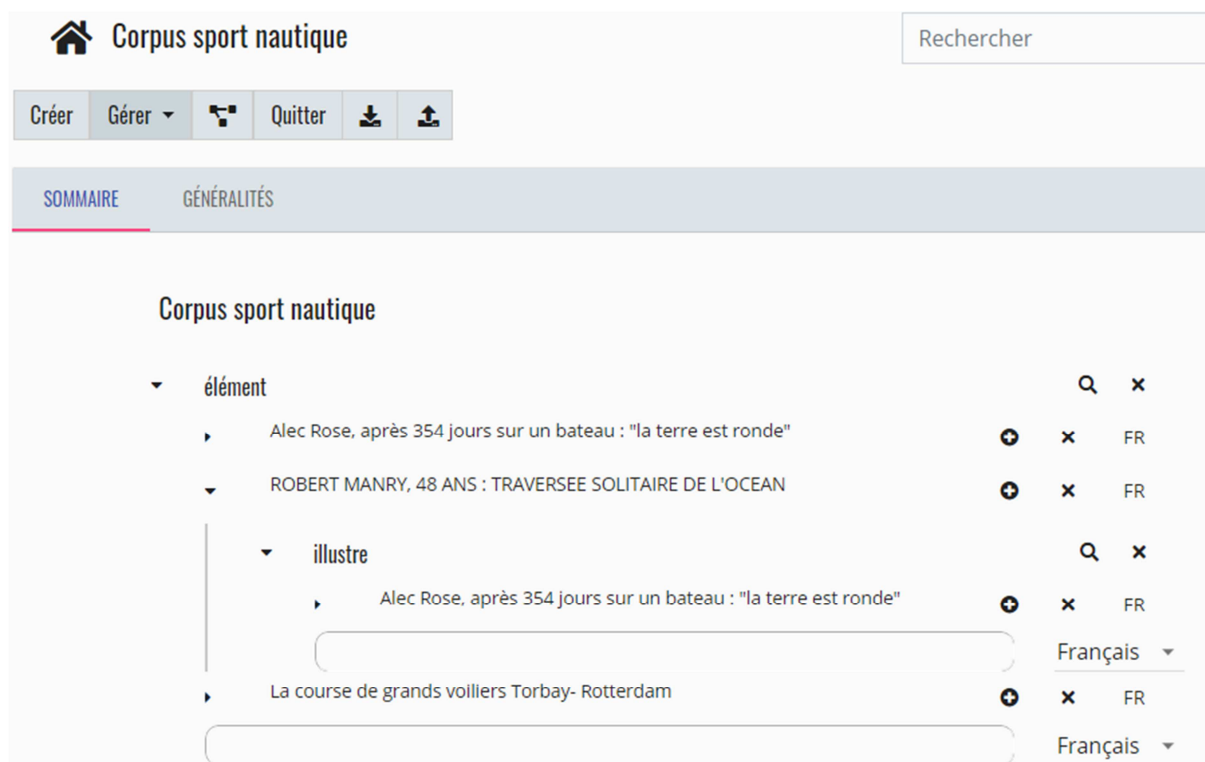


Figure 19. Thematic Corpus “Water Sports”

The Okapi platform exposes a secure SPARQL endpoint and API which allows other ANTRACT tools, especially the TXM platform, to query the knowledge base and to update the stored metadata. For instance, TXM tools could retrieve metadata through the Okapi’s endpoint in order to constitute a corpus. This corpus will then be stored in the knowledge base through the API and used by Okapi to provide thematic publications. Additional semantic descriptors produced by TXM could also be integrated into the Okapi knowledge base.

4. Conclusion

Presented throughout this article, the ANTRACT project’s challenge is to familiarize scholars with the automated research of large audiovisual corpora. Gathering instruments specialized in image, audio and text analysis into a single multimodal apparatus designed to correlate their results, the project intends to develop a transdisciplinary research model suitable to open new perspectives in the study of single or multi-format sources.

At this point of the project, most of the work is dedicated to the development and tuning of the automatic content analysis tools as well as the application of their results to the organization and improvement of the corpus data in connection with research provided by ANTRACT historians (Goetschel, 2019). Their case studies were explored using the TXM textometry platform and the Okapi annotation and publication platform that allows its users to exploit all the data produced by the instruments developed for the project.

From a technological perspective, ANTRACT’s goal is now to further adapt automatic content analysis tools to the specificity of the corpus such as its historical context, its vocabulary, its image format and quality, as it has been done, for instance, by improving the language models used by the automatic transcription tools with the help of the typescripts of voice-overs. Interactive analysis

platforms should also benefit from history scholars feedback in order to improve their user interface and to develop new analytical paths.

At the end of the project, a comprehensive *Les Actualités Françaises* corpus completed with its metadata as well as the results of the research supported by automatic content analysis tools and manual annotations will be made available to the scientific community via the online Okapi platform. To this end, Okapi tutorials will be provided to the public and TXM will continue to be available as an open source software to help the analysis of corpora used in new case studies. Okapi source code will be turned to open source so that other developers can contribute to its enhancement.

Regarding humanities, ANTRACT tools and methodology can be adapted to various types of corpora providing historians as well as specialists from other disciplines such as sociology, anthropology and political science a renewed access to their documents supported by an exhaustive examination of their content.

Acknowledgment

This work has been supported by the French National Research Agency (ANR) within the ANTRACT Project, under grant number ANR-17-CE38-0010, and by the European Union's Horizon 2020 research and innovation program within the MeMAD project (grant agreement No. 780069).

References

- [Ahonen 2006] Ahonen, T., Hadid, A., and Pietikainen, M. "Face description with local binary patterns: Application to face recognition", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28.12 (2006): 2037–2041.
- [Althaus 2018] Althaus, S., Usry, K., Richards, S., Van Thuyle, B., Aron, I., Huang, L., Leetaru, K., Muehlfeld, M., Snouffer, K., Weber, S., Zhang, Y., and Phalen, P. "Global News Broadcasting in the Pre-Television Era: A Cross-National Comparative Analysis of World War II Newsreel Coverage", *Journal of Broadcasting and Electronic Media*, 62.1 (2018): 147-167.
- [Atkinson 2011] Atkinson, N. S., "Newsreels as Domestic Propaganda: Visual Rhetoric at the Dawn of the Cold War", *Rhetoric & Public Affairs*, 14.1 (2011): 69-100.
- [Bartels 2004] Bartels, U. *Die Wochenschau im Dritten Reich. Entwicklung und Funktion eines Massenmediums unter besonderer Berücksichtigung völkisch-nationaler Inhalte*. Peter Lang, Frankfurt am Main (2004).
- [Beloued 2017] Beloued, A., Stockinger, P., and Lalande, S. "Studio Campus AAR: A Semantic Platform for Analyzing and Publishing Audiovisual Corpuses." In *Collective Intelligence and Digital Archives*, John Wiley & Sons Inc., Hoboken, NJ (2017): 85-133.
- [Bewley 2016] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B. "Simple online and realtime tracking". In *IEEE International Conference on Image Processing (ICIP)* (2016): 3464–3468.
- [Bizer 2009] Bizer, C., Heath, T., and Berners-Lee, T. "Linked data - the story so far", *International Journal on Semantic Web and Information Systems*, 5 (2009): 1-22.
- [Bradski 2000] Bradski, G. "The OpenCV Library", *Dr. Dobb's Journal of Software Tools* (2000).
- [Broux 2018] Broux, P.-A., Desnoux, F., Larcher, A., Petitrenaud, S., Carrive, J., and Meignier, S. "S4D: Speaker Diarization Toolkit in Python" *Interspeech*, Hyderabad, India (2018).

- [Cao 2018] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. and Zisserman, A. “Vggface2: A dataset for recognising faces across pose and age”. In 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG) (2018): 67–74.
- [Chambers 2018] Chambers, C., Jönsson, M., and Vande Winkel R. (eds.) *Researching Newsreels. Local, National and Transnational Case Studies*. Global Cinema, Palgrave Macmillan, London (2018).
- [Chenot 2014] Chenot, J.-H., and Daigneault, G. “A large-scale audio and video fingerprints-generated database of TV repeated contents” In 12th International Workshop on Content-Based Multimedia Indexing (CBMI), Klagenfurt, Austria (2014).
- [Christ 1994] Christ, O. “A modular and flexible architecture for an integrated corpus query system” In Ferenc Kiefer et al. (eds.), In 3rd International Conference on Computational Lexicography, Research Institute for Linguistics, Hungarian Academy of Sciences, Budapest (1994): 23-32.
- [Deegan 2012] Deegan, M., and McCarty, W. *Collaborative Research in the Digital Humanities*. Ashgate, Farnham, Burlington (2012).
- [ELAN, 2018] ELAN (Version 5.2) [Computer software]. Max Planck Institute for Psycholinguistics, Nijmegen (2018). Retrieved from <https://tla.mpi.nl/tools/tla-tools/elan>
- [Fein 2004] Fein, S. “New Empire into Old: Making Mexican Newsreels the Cold War Way”, *Diplomatic History*, 28.5 (2004): 703-748.
- [Fein 2008] Fein, S. “Producing the Cold War in Mexico: The Public Limits of Covert Communications” In G. M. Joseph and D. Spenser (eds.), *In from the Cold: Latin America’s New Encounter with the Cold War*, Duke University Press, Durham (2008): 171-213.
- [Feinerer 2008] Feinerer, I., Hornik, K., and Meyer, D. “Text Mining Infrastructure in R”, *Journal of Statistical Software*, 25.5 (2008): 1-54.
- [Goetschel 2011] Goetschel, P., Granger, C. (dir.) “Faire l’événement, un enjeu des sociétés contemporaines”, *Sociétés & Représentations*, 32 (2011): 7-23.
- [Goetschel 2019] Goetschel, P. “Les Actualités Françaises (1945-1969) : le mouvement d’une époque”, #1257, 1 (2019): 34-39.
- [King 2009] King, D. E. “Dlib-ml: A machine learning toolkit”. *Journal of Machine Learning Research*, 10 (2009): 1755–1758.
- [Heiden 2010] Heiden, S. “The TXM Platform: Building Open-Source Textual Analysis Software Compatible with the TEI Encoding Scheme” In R. Otaguro, K. Ishikawa, H. Umemoto, K. Yoshimoto, Y. Harada (eds.), *24th Pacific Asia Conference on Language, Information and Computation*, Institute for Digital Enhancement of Cognitive Development, Waseda University (2010).
- [Hotho 2005] Hotho, A., Nürnberger, A. and Paaß, G. “A brief survey of text mining”, *LDV Forum*, 20.1 (2005): 19-62.
- [Imesch 2016] Imesch, K., Schade, S., Sieber, S. (eds.) *Constructions of cultural identities in newsreel cinema and television after 1945*. MediaAnalysis, 17, transcript-Verlag, Bielefeld (2016).
- [Lebart 1998] Lebart, L., Salem, A. and Berry, L. *Exploring textual data. Text, speech, and language technology*, 4, Kluwer Academic, Dordrecht, Boston (1998).
- [Lindeperg 2000] Lindeperg, S. *Clio de 5 à 7 : les actualités filmées à la Libération*, archive du futur. CNRS, Paris (2000).

- [Lindeperg 2008] Lindeperg, S. “Spectacles du pouvoir gaullien: le rendez-vous manqué des actualités filmées”. In J.-P. Bertin-Maghit (dir.), *Une histoire mondiale des cinémas de propagande*, Nouveau Monde Éditions, Paris (2008): 497-511.
- [MacWhinney, 2000] McWhinney, B. *The CHILDES Project: Tools for Analyzing Talk*. L. Erlbaum Associates, Mahwah, N.J. (2000).
- [Maitland 2015] Maitland, S. “Culture in translation: The case of British Pathé News” In *Culture and news translation, Perspectives: Studies in Translation Theory and Practice*, 23.4 (2015): 570-585.
- [McEnery 2012] McEnery, T. and Hardie, A. *Corpus linguistics: method, theory and practice*. Cambridge University Press, Cambridge (2012).
- [Motik 2012] Motik, B., Patel-Schneider, P. F., Parsia, B. “OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition)”. W3C Recommendation (2012).
- [Pincemin 2020] Pincemin, B., Heiden, S. and Decorde, M. “Textometry on Audiovisual Corpora. Experiments with TXM software”, 15th International Conference on Statistical Analysis of Textual Data (JADT), Toulouse (2020).
- [Povey 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G. and Vesely, K. “The kaldi speech recognition toolkit” In *IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, Hilton Waikoloa Village, Big Island, Hawaii, US (2011).
- [Povey 2016] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. “Purely sequence-trained neural networks for ASR based on lattice-free MMI” *Interspeech*, San Francisco (2016): 2751–2755.
- [Pozner 2008] Pozner, V. “Les actualités soviétiques durant la Seconde Guerre mondiale : nouvelles sources, nouvelles approches” In J.-P. Bertin-Maghit (dir.), *Une histoire mondiale des cinémas de propagande*, Nouveau Monde Editions, Paris (2008): 421-444.
- [R Core Team 2014] R Core Team., “R: A Language and Environment for Statistical Computing”, R Foundation for Statistical Computing, Vienna, Austria (2014).
- [Salem 2004] Salem, A. “Introduction à la résonance textuelle” In G. Purnelle et al. (eds.), *7èmes Journées internationales d’Analyse statistique des Données Textuelles*, Presses universitaires de Louvain, Louvain (2004): 986–992.
- [Schmid 1994] Schmid, H. “Probabilistic Part-of-Speech Tagging Using Decision Trees” In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK (1994).
- [Schroff 2015] Schroff, F., Kalenichenko, D. and Philbin, J. “Facenet: A unified embedding for face recognition and clustering”. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015): 815–823.
- [Szegedy 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. “Rethinking the Inception Architecture for Computer Vision” In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas (2016).
- [Troncy 2012] Troncy, R., Mannens, E., Pfeiffer, S. and van Deursen, D. “Media Fragments URI 1.0 (basic)”. W3C Recommendation (2012).
- [Veray 1995] Veray, L. *Les Films d’actualités français de la Grande Guerre*. SIRPA/AFRHC, Paris (1995).
- [Viola 2004] Viola, P. and Jones, M. J. “Robust real-time face detection”. *International Journal of Computer Vision*, 57.2 (2004): 137–154.
- [Weiss 2015] Weiss, S. M., Indurkha, N., and Zhang, T. *Fundamentals of Predictive Text Mining*. Springer-Verlag, London (2015).

[Zhang 2016] Zhang, K., Zhang, Z., Li, Z. and Qiao, Y. “Joint face detection and alignment using multitask cascaded convolutional networks”. IEEE Signal Processing Letters, 23.10 (2016): 1499–1503.

¹ ANTRACT: ANalyse TRansdisciplinaire des ACTualités filmées (1945-1969).

² ESTER 1 & 2, EPAC, ETAPE, and REPERE corpus available in ELRA catalogues (<http://www.elra.info/>).

³ ETAPE, and QUAERO corpus available in ELRA catalogues (<http://www.elra.info/>).

⁴ <https://github.com/XuezheMax/NeuroNLP2>

⁵ Challenge REPERE, test data.

⁶ <https://github.com/Linzaer/Face-Track-Detect-Extract>

⁷ Text Encoding Initiative, <https://tei-c.org>


B.9 EURECOM's LDK 2021 conference paper

This paper describes the ZeSTE approach developed by EURECOM and accepted at LDK 2021.

1 Explainable Zero-Shot Topic Extraction Using a 2 Common-Sense Knowledge Graph

3 Ismail Harrando ✉ 

4 EURECOM, Sophia Antipolis, Biot, France

5 Raphaël Troncy ✉ 

6 EURECOM, Sophia Antipolis, Biot, France

7 — Abstract —

8 Pre-trained word embeddings constitute an essential building block for many NLP systems and
9 applications, notably when labeled data is scarce. However, since they compress word meanings into a
10 fixed-dimensional representation, their use usually lack interpretability beyond a measure of similarity
11 and linear analogies that do not always reflect real-world word relatedness, which can be important for
12 many NLP applications. In this paper, we propose a model which extracts topics from text documents
13 based on the common-sense knowledge available in ConceptNet [24] – a semantic concept graph that
14 explicitly encodes real-world relations between words – and without any human supervision. When
15 combining both ConceptNet’s knowledge graph and graph embeddings, our approach outperforms
16 other baselines in the zero-shot setting, while generating a human-understandable explanation for
17 its predictions. We study the importance of some modeling choices and criteria for designing the
18 model, and we demonstrate that it can be used to label data for a supervised classifier to achieve an
19 even better performance without relying on any humanly-annotated training data. We publish the
20 code of our approach at <https://github.com/D2KLab/ZeSTE> and we provide a user friendly demo at
21 <https://zeste.tools.eurecom.fr/>.

22 **2012 ACM Subject Classification** Computing methodologies → Information extraction

23 **Keywords and phrases** Topic Extraction, Zero-Shot Classification, Explainable NLP, Knowledge
24 Graph

25 **Digital Object Identifier** 10.4230/OASICS.CVIT.2016.23

26 **Supplementary Material** Source code at <http://github.com/D2KLab/ZeSTE>

27 **Funding** This work has been partially supported by the French National Research Agency (ANR)
28 within the ASRAEL (ANR-15-CE23-0018) and ANTRACT (ANR-17-CE38-0010) projects, and
29 by the European Union’s Horizon 2020 research and innovation program within the MeMAD (GA
30 780069) and SILKNOW (GA 769504) projects.

31 **1 Introduction**

32 Word2Vec [14], GloVe [16], BERT [5] along with its many variants are among the most
33 cited works in NLP. They have demonstrated the possibility of creating generic, cross-task,
34 context-free and contextualized word representations from big volumes of unlabeled text,
35 which can be then used to improve the performance of numerous down-stream NLP tasks
36 by bringing free “real world knowledge” about words meanings and usage, learned mostly
37 through word co-occurrences statistics, thus cutting down the need for substantial amounts of
38 labeled data. However, being compacted representations of word meanings, these embeddings
39 do not offer much in terms of interpretation: we know that similar words tend to have
40 similar representations (i.e. similar orientation in the embedding space), and that some
41 analogies can be found by doing linear algebraic operations in the embedding space (such
42 as the now-famous $v_{King} - v_{Man} + v_{Woman} \approx v_{Queen}$). Both measures, however, fall short
43 when evaluated systematically, as there is an entire literature about studying the limits of



© Ismail Harrando and Raphaël Troncy;
licensed under Creative Commons License CC-BY 4.0
42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:15



OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

analogies and the biases that these word embeddings can encode depending on the corpora they have been trained on [4, 2, 15, 13].

In this paper, we consider the task of *topic categorization*, a sub-task of text classification where the goal is to label a textual document such as a news article or a video transcript, into one of multiple predefined *topics*, i.e. labels that are related to the topical content of the document. Common examples for news topics are “*Politics*”, “*Sports*” and “*Business*”. What is interesting about this task, compared to other text classification tasks such as *spam detection* or *sentiment analysis*, is that the content of the document to classify is *semantically related* to the labels themselves, providing an interesting case for zero-shot prediction setting. Zero-shot prediction, broadly defined, is the task of predicting the class for some input without having been exposed to any labeled data from that class.

To do so, we propose to leverage *ConceptNet*, a knowledge graph that aims to model common sense knowledge into a computer- and human-readable formalism. Coupled with its graph embeddings (ConceptNet Numberbatch¹), we show that using this resource does not only achieve better empirical results on the task of zero-shot topic categorization, but also does so in an explainable fashion. With every word being a node in the knowledge graph, it is straightforward to justify the similarity between words in the document and its assigned label, which is not possible for other distributional word embeddings as they are built on the statistical aggregations of large volumes of textual data.

The remainder of this paper is structured as follows: we present some related work for text categorization emphasizing the methods that make use of external semantic knowledge (Section 2). We present our proposed method, named **ZeSTE** (**Z**ero **S**hot **T**opic **E**xtraction) in Section 3. We empirically evaluate our approach for zero-shot topic categorization in Section 4 where we compare it to different baselines on multiple topic categorization benchmark datasets (including a non-English dataset). We also test our method against a few-shot setup and show how our approach can be combined with a supervised classifier to obtain competitive results on the studied datasets without relying on any annotated data. In Section 5, we describe a demo that we developed that enable users to provide their own set of labels and observe the explanations for the model predictions. Finally, we conclude and outline some potential future improvements in Section 6.

2 Related Work

Nearly all recent state-of-the-art Text Categorization models ([29, 3, 28, 25], to cite a few) rely on some form of Transformer-based architecture [27], pre-trained on large text corpora. While the task of using fully-unsupervised, non-parametric models for text categorization is yet to be explored to the best of our knowledge, there has been multiple efforts to incorporate common-sense knowledge as a basis for many artificial intelligence tasks, especially in a zero-shot setting where humans seem to be able to satisfactorily perform a new task by relying mostly on their common sense and prior knowledge accumulated from their interaction with the world.

In this paper, we propose to leverage ConceptNet [24], a multilingual semantic graph containing statements about common-sense knowledge. The nodes represent concepts (words and phrases, e.g. `/c/en/sport`, `/c/en/belief_system`, `/c/en/ideology`, `/c/fr/coup_d'état`) from 78 languages, linked together by semantic relations such as `/r/IsA`, `/r/RelatedTo`, `/r/Synonym`, `/r/PartOf`. The graph contains over 8 million nodes and 21 million edges,

¹ <https://github.com/commonsense/conceptnet-numberbatch>

expressed in triplets such as (`/c/en/president`, `/r/DefinedAs`, `/c/en/head_of_state`). It was built by aggregating facts from the Open Mind Common Sense project [20], parsing Wiktionary², Multilingual WordNet [8], OpenCyc [7], as well as a subset of DBpedia, and designed to explicitly express facts about the real world and the usage of words and concepts that is necessary to understand natural language. Along with the graph, *ConceptNet Numberbatch* are multilingual pre-trained word (and concept) embeddings that are built on top of the ConceptNet knowledge graph. They are generated by computing the Positive Pointwise Mutual Information (PPMI) for the matrix representation of the graph, reducing its dimensionality, and then using “expanded retrofitting” [23] to make them more robust and linguistically representative by combining them with Word2Vec and GloVe embeddings. While the approach can be carried using other linguistic resources such as WordNet [8], we choose to use ConceptNet because it models word relations that are more relevant to the task of Topic Categorization such as `/r/RelatedTo`, which is the most present relation in the graph.

[6] is an early example of leveraging semantic knowledge to improve text categorization. It uses the relations in WordNet [8] to enhance the Bag of Word representation of documents by mapping the different words from a document into their entries in WordNet, and adding those as well as their hypernyms to the Bag of Words count. This, followed by a statistical χ^2 test to reduce the dimension of the feature vector, leads to a significant improvement over the simple bag-of-words model. [21] introduces *Graph of Words*, in which every document is represented by a graph of its terms, all connected with relations reflecting the co-occurrence information (terms appearing within a window of size w are joined by an edge). The authors propose a weighting scheme for the traditional TF-IDF model, where nodes are weighted based on some graph centrality measure (degree, closeness, PageRank), and edges are weighted with Word2Vec word embedding cosine similarity between their nodes. Incorporating both graph structure and distributional semantics from the embeddings to compute a weight for each term yields significantly better results on multiple text classification datasets.

[30] benchmark the task of zero-shot text classification, underlining the lack of work reported on this challenge in the NLP community in comparison to the field of computer vision. They distinguish two definitions of zero-shot text categorization: *Restrictive*, in which during a training phase, the classifier is allowed to see a subset of the data with the corresponding labels, but during inference, it is tested on a new subset of examples from the same dataset but not pertaining to any of the seen labels; *Wild*, where the classifier is not allowed to see any examples from the labeled data but can use Wikipedia’s categories as a proxy dataset, for example. Our method fits into this second definition, although it does not require any training data. The authors compare some methods in both regimes (restrictive and wild) and they propose “Entail”, a model based on BERT [5] and trained on the task of textual entailment evaluated on the Yahoo! Comprehensive Questions and Answers dataset.

[17] tackle the task of zero-shot text classification by projecting both the document and the label into an embedding space and using multiple architectures to measure the relatedness of the document and label embeddings. At test time, the classifier is able to ingest labels that were not seen during the training phase, but share the same embedding space with the labels already seen. A similar approach is followed by [22], in which both documents and labels are embedded into a shared cross-lingual semantic representations (CLESA) built upon Wikipedia as a multilingual corpus, and then the prediction is made by measuring the similarity between the two representations.

² https://en.wiktionary.org/wiki/Wiktionary:Main_Page

Finally, [31] propose a two-stage framework for zero-shot document categorization, combining 4 kinds of semantic knowledge: distributional word embeddings, class descriptions, class hierarchy, and the ConceptNet knowledge graph. In the first phase, a (coarse-grained) classifier is trained to decide whether the document at hand comes from a class that was seen during the training phase or not. This is done by training one ConvNet classifier [11] per label in the “seen” dataset, and setting a confidence threshold that, if none of the classifiers meets, the document is considered to be for the unseen labels. Secondly, a fine-grained classifier predicts the document final label. If the document is from a “seen” label, then the corresponding pretrained ConvNet classifier is picked. Otherwise, a zero-shot classifier which takes as input a representation of the document, the label, and their ConceptNet closeness, is trained on the seen labels but is expected to generalize to unseen ones as they share the same embedding space.

3 Approach

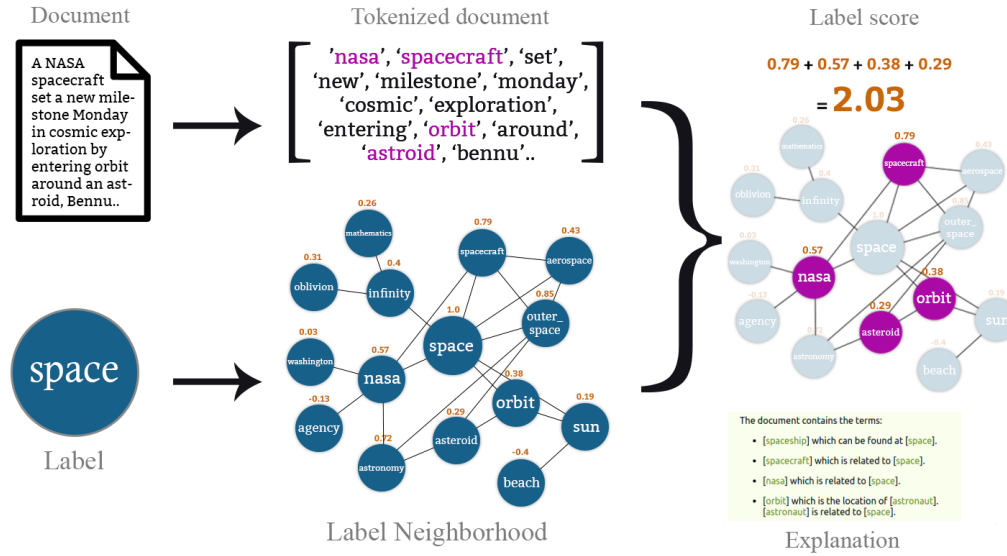
Our approach aims to perform topic categorization without relying on any in-domain labeled or unlabeled examples. Our underlying assumption is that words belonging to a certain topic are part of a vocabulary that is semantically related to its humanly-selected candidate label, e.g. a document about the topic of “*Sports*” will likely mention words that are semantically related to the word *Sport* itself, such as *team*, *ball*, and *score*. We use ConceptNet [24] to produce a list of candidate words related to the labels we are interested in. We generate a “topic neighborhood” for each topic label which contains all the semantically related concepts/nodes, and we then compute a score for each label based on the document content. Figure 1 illustrates our approach using a simple example.

3.1 Generating Topic Neighborhoods

To generate the topic neighborhoods for a given label, we query ConceptNet for nodes that are directly connected to the label node. Since the number of calls to the online API is capped at 120 queries/minute, we instead use the dump³ of all ConceptNet v5.7 assertions, keeping only the English and French concepts for the English and French datasets, resulting in 3,323,321 (resp. 2,943,446) triplets, respectively. Although the assertions contain a finer granularity when it comes to referring to concepts, we only consider the root word for each concept to build the neighborhood. For example, the word “match” has multiple meanings: the tool to light a fire /c/en/match/n/wn/artifact, the event where two contenders meet to play /c/en/match/n/wn/event, and the concept of several things fitting together /c/en/match/n/wn/cognition. All these nodes (as well as others such as the verb form) will be mapped to the same term: “match”. We also add (inverse) relations from the object to the subject for each triplet to ensure that every term in the graph has a neighborhood. The total number of unique triplets is 6,412,966, with 1,165,189 unique nodes for English (6,413,002 and 1,448,297 for French, respectively).

The topic neighborhood is created by querying every node that is N hops away from the label node. Every node is then given a score that is based on the cosine similarity between the label and the node computed using *ConceptNet Numberbatch* (ConceptNet’s graph embeddings). This score represents the relevance of any term in the neighborhood to the main label, and would also allow us to refine the neighborhood and produce a score. In the

³ <https://github.com/commonsense/conceptnet5/wiki/Downloads#assertions>



■ **Figure 1** Illustration of ZeSTE: given a document and a label, we start by pre-processing and tokenizing the document into a list of terms, and we generate the label neighborhood graph by querying ConceptNet (we omit relation labels in the figure for clarity). Each node on the graph is associated with a score that corresponds to the cosine similarity between the graph embeddings of that node and the label node. We use the overlap between the document terms and the label neighborhood to generate a score for the label, as well as an explanation for the prediction. After doing so for all candidate labels, we pick the one with the highest score to associate to the document at hand.

case of a label which has multiple tokens (e.g. the topic “Arts, Culture, and Entertainment”), we just take the union of all word components’ neighborhoods, weighted by the maximum similarity score if the same concept appear in the vicinity of multiple label components.

The higher N is, and the bigger the generated neighborhoods become. We thus propose multiple methods to vary the size of the neighborhood:

1. **Coverage:** we vary the number of hops N ;
2. **Relation masking:** we consider subsets of all possible relations between words from the ConceptNet knowledge graph. More precisely, we consider three cases:
 - a. The sole relation *RelatedTo* which is the most frequent one in the graph;
 - b. The 10 semantic and lexical *similarity* relations only, i.e. *DefinedAs*, *DerivedFrom*, *HasA*, *InstanceOf*, *IsA*, *PartOf*, *RelatedTo*, *SimilarTo*, *Synonym*, *Antonym*;
 - c. The whole set of 47 relations defined in ConceptNet.
3. **Filtering:** we filter out some nodes based on their similarity score:
 - a. Threshold (*Thresh T*): we only keep nodes in the neighborhood if their similarity score to the label node is greater than a given threshold T .
 - b. Hard Cut (*Top N*): we only keep the top N nodes in the neighborhood ranked by their similarity score.
 - c. Soft Cut (*Top P%*): we only keep the top $P\%$ nodes in the neighborhood, ranked on their similarity score.

3.2 Scoring a Document

Once the neighborhood is generated, we can predict the document label by quantifying the overlap between the document content (as broken down to a list of tokens) and the label neighborhood nodes, which we denote in the following equations as $doc \cap LN(label)$. We consider the following scoring schemes:

1. **Counting**: assigning the document with the highest overlap count between its terms and the topic neighborhood.

$$count_score(doc, label) = |doc \cap LN(label)| \quad (1)$$

2. **Distance**: factoring in the graph the distance between the term in the document and the label (number of nodes or path length between the token node and the label): the further a term is from the label vicinity, the lower is its contribution to the score.

$$distance_score(doc, label) = \sum_{token \in doc \cap LN(label)} \frac{1}{min_path_length(token, label) + 1} \quad (2)$$

3. **Degree**: each node's score is computed using the number of incoming edges to it, reflecting its importance in the topic graph (we use $f(n) = \log(1 + n_{edges})$ to amortize nodes with a very high degree).

$$degree_score(doc, label) = \sum_{token \in doc \cap LN(label)} f(node_degree(token)) \quad (3)$$

4. **Numberbatch similarity**: for each term in the document included in the label neighborhood, we increase the score by its similarity to the label embedding (we denote the Numberbatch concept embedding for word w by nb_w).

$$numberbatch_score(doc, label) = \sum_{token \in doc \cap LN(label)} sim(nb_{token}, nb_{label}) \quad (4)$$

5. **Word Embedding similarity**: similar to the Numberbatch similarity, but we use pre-trained 300-dimensional GloVe [16] word embeddings instead to measure the word similarity (we denote the GloVe word embedding for word w by $glove_w$).

$$glove_score(doc, label) = \sum_{token \in doc \cap LN(label)} sim(glove_{token}, glove_{label}) \quad (5)$$

We observe that in equations 4 and 5, multiple similarity measures and normalization options were considered, but the cosine similarity empirically showed the best results, so it has been used for the rest of the experiments. The model is thus the set of the neighborhood for each candidate label coupled with a scoring scheme. We discuss in Section 4.2 (Model Selection) how to empirically decide on the best filtering and scoring method that we then use in our experiments and our online demo.

3.3 Explainability

Given the label neighborhood, we can generate an explanation as to why a document has been given a specific label. This explanation can be generated in Natural Language or shown

as the subgraph of ConceptNet that connects the label node and every word in the document that appears within its neighborhood, and hence counted towards its score^{3.1}. Since this graph is usually quite big, we can generate a more manageable summary by picking up the closest N terms to the label in the graph embedding space, as they constitute the nodes contributing most to the score of the document. We can show one path (for instance, the shortest) between each of the top term nodes and the label node. The paths can then be verbalized in natural language. For example, for the label **Sport**, and a document containing the word *Stadium*, a line from the explanation (i.e. a path on the explanation subgraph) would look like this (**r/RelatedTo** and **r/IsA** are two relations from ConceptNet):

The document contains the word “Stadium”, which is *related to* “Baseball”. “Baseball” is a “Sport”.

Another method of explaining the predictions of the model is to highlight the words (or n-grams) that contributed to the classification score in the document. Since every word that appear both in the document and the label neighborhood has a similarity score associated to it (e.g. the cosine similarity between the word and the label embedding), we can visually highlight the words that are relevant to the topic. These two explanation methods are further discussed in the Section 5.

4 Experiments

In this section, we first describe the datasets which have been used to evaluate our approach (Section 4.1). Next, we present experiments to select the best model (Section 4.2). We then detail the zero-shot baselines that we compare to our approach (Section 4.3) before discussing our results (Section 4.4). Finally, we show how our model can be used to bootstrap the training for supervised classifier to achieve significantly better results (Section 4.5).

4.1 Datasets

While the premise of our approach is the possibility to perform topic categorization in a zero-shot setting, we evaluate it on several datasets from the literature. We identify 4 different Topic Categorization datasets with different properties in terms of style (professional news sources or user-generated content), size, number of topics, topic distribution and document length. We also evaluate our model on a new dataset named AFP News, which provides interesting comparison grounds such as multilingualism (available in English and French), multi-topical documents and strong imbalance in topics distribution. Table 3 summarizes the characteristics of each of these 5 datasets.

- **20 Newsgroups** [12]: a collection of 18000 user-generated forum posts arranged into 20 groups seen as topics such as “Baseball”, “Space”, “Cryptography”, and “Middle East”.
- **AFP News** [18]: a dataset containing 125K English and 26K French news articles issued by the French News Agency (*Agence France Presse*). The articles are tagged with one or more topics coming from IPTC NewsCode taxonomy⁴. We consider the first level of this taxonomy which corresponds to 17 top-level topics such as “Art, Culture and Entertainment”, “Environment”, or “Lifestyle and Leisure”. The label distribution is highly unbalanced. Since the data on both the English and French documents come from

⁴ <http://cv.iptc.org/newscodes/subjectcode/>

the same source and have similar properties, we use this dataset to compare how well our method compare on two different languages.

- **AG News** [10]: a news dataset containing 127600 English news articles from various sources. Articles are fairly distributed among 4 categories: “*World*”, “*Sports*”, “*Business*” and “*Sci/Tech*”.
- **BBC News** [9]: a news dataset from BBC containing 2225 English news articles classified in 5 categories: “*Politics*”, “*Business*”, “*Entertainment*”, “*Sports*” and “*Tech*”.
- **Yahoo! Answers Comprehensive Dataset** [26]: a dataset containing over 4 million questions (title and body) and their answers submitted by users, extracted from the Yahoo! Answers website. We construct the evaluation dataset following the procedure described in [30] to reproduce its setup for comparison: we select 10K questions from each of the top 10 categories on Yahoo! Answers. We split it into 2 categories. The first split contains the labels “*Health*”, “*Family & Relationships*”, “*Business & Finance*”, “*Computers and Internet*” and “*Society and Culture*” whereas the second split contains the labels “*Entertainment & Music*”, “*Sports*”, “*Science & Mathematics*”, “*Education & Reference*”, and “*Politics & Government*”. The ground-truth topic labels are assigned by users.

In order to determine the filtering criteria as discussed in Section 4.2 without relying on any further dataset-specific tuning, we use the BBC News dataset as a development set to select the optimal parameters for our model, under the hypothesis that the properties that work best for this dataset would work best for others as well. We verify post-hoc that this hypothesis holds empirically, i.e., the design choices decided using BBC News turn out to deliver the best results on the other datasets as well. The filtering criteria values that gave the best results for *Threshold*, *Hard Cut* and *Soft Cut* have empirically been set to $T = 0.0$, $N = 20000$, $P = 50\%$, respectively.

The 5 datasets have all been pre-processed using the same procedure: we lowercase the text, remove all non-alphabetical symbols and English (or French) stopwords. We then tokenize the strings using the space as separator and finally lemmatize the word using `WordNetLemmatizer`⁵. If the dataset has multiple textual contents (e.g. the Yahoo! Questions dataset consists of questions that are made of a title, a question body, and a set of answers), we concatenate them to form one “document”. In the case of the AFP News dataset, each document can be tagged with one label, multiple labels, or no labels. We drop all non-tagged documents. To compute accuracy, we consider a prediction to be correct if it is among the document labels, and false otherwise. Finally, for the 20 Newsgroups dataset, we collapse the categories “comp.os.ms-windows.misc” and “comp.windows.x” into “windows”, and “comp.sys.mac.hardware” and “comp.sys.ibm.pc.hardware” into “hardware”, since they have very similar original labels. We do so for the baselines methods as well.

4.2 Model Selection

In this section, we evaluate some of the options regarding the neighborhood filtering and document scoring mentioned in Section 3. We use the *BBC News* dataset as a testbed for evaluating model selection. We report the results on the other datasets using the best parameters found at this stage. We first evaluate the different choices made to generate the label neighborhood as discussed in Section 3.1 and reported in Table 1.

⁵ <http://www.nltk.org/api/nltk.stem.html?highlight=lemmatizer#module-nltk.stem.wordnet>

Relations	Depth	Filtering method			
		Keep All	Top50%	Top20K	Thresh
One	N = 1	55.4	54.5	55.4	55.4
	N = 2	69.0	65.8	64.8	66.2
	N = 3	81.0	81.3	83.5	81.3
Similarity	N = 1	60.8	57.5	60.8	60.8
	N = 2	70.3	66.9	66.2	68.0
	N = 3	77.9	81.9	83.4	81.9
All	N = 1	68.4	67.4	68.4	68.4
	N = 2	75.2	73.8	78.0	73.9
	N = 3	83.6	83.6	84.0	83.6

■ **Table 1** Comparing the different filtering configurations on the BBC News dataset (performance expressed in Accuracy)

We observe that the most consistent way of improving the results is to use larger neighborhoods, as 3-hops neighborhoods systematically outperform the 1 and 2-hops ones. Our experiments show that going beyond $N = 3$ comes at the cost of increasing the computation time (mainly the computation of cosine similarity between the label and related nodes), while offering only very marginal improvement overall. The filtering method also impacts the performance but not as consistently (especially for $N = 3$). Finally, using all the relations generally yields better results than using only a subset of the relations, enough to justify the speed trade-off. It is also worth noting that using only the “r/RelatedTo” relation yields comparatively good results, which highlights the fact that “common-sense word relatedness” as expressed in ConceptNet is a strong signal for topic categorization.

For the scoring scheme, we evaluate the various methods mentioned in Section 3.2. The results are reported in Table 2.

Count	Distance	Degree	Numberbatch	GloVe
81.8	77.8	78.1	84.0	81.6

■ **Table 2** Evaluating the scoring schemes on BBC News (performance expressed in Accuracy)

We see that using the ConceptNet Numberbatch embeddings gives the best result as they can condense the count, distance, degree of the nodes and the linguistic similarity with regard to the label into a measure of similarity in the embedding space. Accounting for term frequency (counting a word twice in the scoring if it appears twice in the document) in all of the scoring schemes did not translate to an improvement on the results. Accounting for n-grams, however, seems to slightly improve the results, but they require the availability of a corpus to mine such n-grams. Therefore, for the rest of our experiments, we do not account for n-grams. For the rest of our experiments, we keep the following configuration: (*'All relations', $N = 3$, 'Top20K', 'Numberbatch scoring'*). We use ConceptNet v5.7 and Numberbatch embeddings v19.08.

4.3 Baselines

We propose 3 baseline systems:

Dataset	BBC News	AG News	20 Newsgroups	AFP News (FR)	YQA-v0	YQA-v1
# topics	5	4	20	17	5	5
# docs	2225	127600	18000	125516	50000	50000
doc/topic std	54.3	22.4	56.7	13682.7	0.0	0.0
Avg. words/doc	390	40	122	242	43	44
EN	26.1	26.7	53.5	60.0	51.8	36.2
GWA	40.2	63.9	36.7	32.8	49.9	43.4
Entail [30]	71.1	64.0	45.8	61.8	52.0	49.3
ZeSTE	84.0	72.0	63.0	80.9 (78.2)	60.3	58.4
Supervised	96.4	95.5	88.5		72.6	80.6
Method	[19]	[29]	[28]			[30]

■ **Table 3** Performance on five Topic Categorization datasets (Accuracy)

- 337 ■ *Entail*: this model is provided by HuggingFace⁶ [30]. We use `bart-large-mnli` as
338 our backend Transformer model which can also be tested at [https://huggingface.co/](https://huggingface.co/zero-shot/)
339 [zero-shot/](https://huggingface.co/zero-shot/).
- 340 ■ *GloVe Weighted Average* (GWA) inspired by [1]: we average the 300-d GloVe embeddings
341 vectors for every word in the document, and use the cosine similarity between the
342 document embedding and the GloVe label embedding as a score to classify the document.
343 For multi-worded labels (e.g. “Middle East”), we use the average vector of all the label
344 components as the label embedding.
- 345 ■ *Embedding Neighborhood* (EN): for each label, we select the 20k closest words in the
346 embedding space. We score each document by adding up the cosine similarity between
347 the GloVe embedding of every word in the document that appears in the “embedding
348 neighborhood” and the GloVe embedding of the label. In other words, we substitute the
349 explicit graph connections in ConceptNet with the closeness in the GloVe embedding
350 space. This baseline reflects the ability of generic embeddings to encode the topicality of
351 words based only on the similarity in the embedding space.

352 4.4 Zero-Shot Results

353 We provide the results obtained by evaluating our method against the baselines on the 5
354 datasets (BBC News, AG News, 20 Newsgroups, AFP News and YQA) in Table 3. Our
355 method surpasses both GloVe baselines with a significant margin in accuracy on all datasets.
356 GWA shows that the generic word embeddings poorly encode the topicality of words, as it
357 is based solely on the similarity scores between the document content and the label world
358 embedding. The low results with EN show that filtering based only on the embedding space
359 (instead of the graph) is insufficient since the rarely-used words tend to clutter the embedding
360 neighborhood. ZeSTE significantly outperforms Entail, despite the fact that the later relies
361 on a large corpus pre-training and *textual entailment* task fine-tuning.

362 The confusion matrices for each datasets (Figure 2) indicate that our method performs
363 more poorly on datasets where there is a lot of topical overlap between the different labels.
364 For example, on 20 Newsgroups, “alt.atheism”, “soc.religion.christian”, “talk.religion.misc”
365 have a lot of overlapping vocabulary, leading to most documents under “alt.atheism” to

⁶ We are using the implementation provided at <https://github.com/katanaml/sample-apps/tree/master/01>

fall into either other options. If we collapse all three labels into one (e.g. “religion”), the performance improves from 63.0% to 68.9%. We also observe on the AFP News dataset that “politics” intersects with “unrest, conflict, war” and “business, finance”. The lack of a diameter pattern in AFP’s confusion matrix is due to the high imbalance in the labels, which hurts the precision of the model. It is also worth mentioning how the method works seamlessly for other languages, as demonstrated on the French AFP News dataset, which sees a slight drop of accuracy from 80.9% on English to 78.2% accuracy on French. This shows a great potential for multilingual applicability as ConceptNet supports 78 languages.

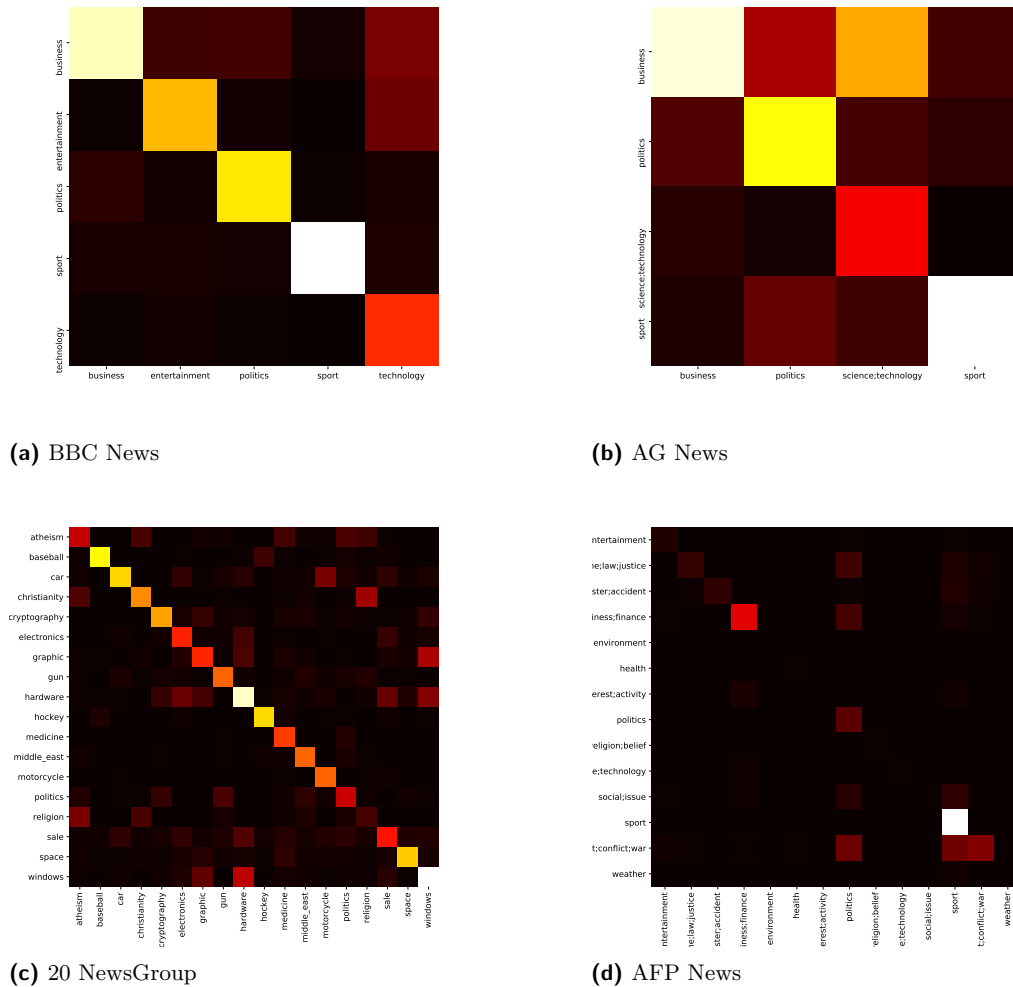


Figure 2 Confusion Matrices for the 4 news datasets

Our method is clearly outperformed by the fully supervised methods. While the drop in performance is significant for some datasets, it is to be observed that the supervised methods not only rely on the availability of labeled training data, but usually also require expensive pre-training on more data. For instance, [29] use XLNet, an autoregressive Transformer that has been pre-trained on 120 GB of text. We consider that this absolute loss of accuracy performance is counter-balanced by the applicability in a zero-shot setting as well as the explainability of the model’s decision.

Finally, we note that the choice of the initial label can be critical for the functioning of this method. While we stayed true to the original labels in the experiments (with an exception for the label “World” that was replaced with “news, politics” in the AG News dataset), we are aware of the possibility of obtaining even better results by changing a label to a more fitting one or including more keywords into it.

4.5 Few-Shots Setup

For each dataset, we compare our model to a more realistic use-case. We create a 80-20 training/test split if one is not already provided, and we randomly sample n examples from each category to create a training set for our supervised classifier. Among the classifiers considered, we find uncased BERT (*BertForSequenceClassification*) to perform the best. We grow n in increments of 10 until we achieve an empirical accuracy score on the test set that surpasses our approach in the zero-shot setting. We report $N = n * |labels|$ the number of documents that need to be annotated in Table 4. We also observe that increasing the number of documents does not always improve the test set accuracy.

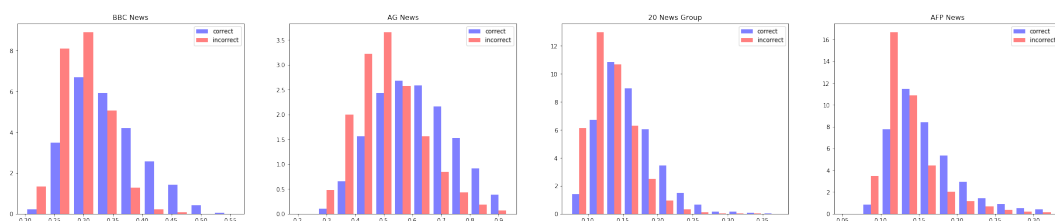
Dataset	BBC News	AG News	20 Newsgroups	AFP News
N	300	240	2160	8500

■ **Table 4** The required number of documents needed to achieve zero-shot best performance

4.6 Bootstrapping a Supervised Classifier

One of the potential usage of zero-shot classification is to provide “automatic labeling” for unlabeled documents to a traditional supervised classifier. In other words, we use ZeSTE to annotate a portion of each dataset, and we feed these annotated examples to a state-of-the-art text classifier.

We first define the confidence of the classification as the normalized score for each label, i.e. divided by the sum of all candidate labels scores. In Figure 3, which shows the error distribution with respect to the classification confidence, we see that it correlates well with whether the label is correct or not. Therefore, we can use it as a signal to pick samples to use to bootstrap our classifier. We train the same few-shots model from 4.5 on the best 60% examples of our training data, i.e. we drop 40% of the training examples on which ZeSTE is least confident. We report on the results in Table 5 (the results for ZeSTE row correspond to the performance on the test-set only, not the entire dataset as in Table 3). We can clearly see how the bootstrapping process helps the classifier achieving significantly better results on all tested datasets, all without requiring any human annotation. It is worth mentioning that for this application, the BERT-based classifier training was not thoroughly fine-tuned, which means that even better results can be achieved using the same automatic labeling setup.



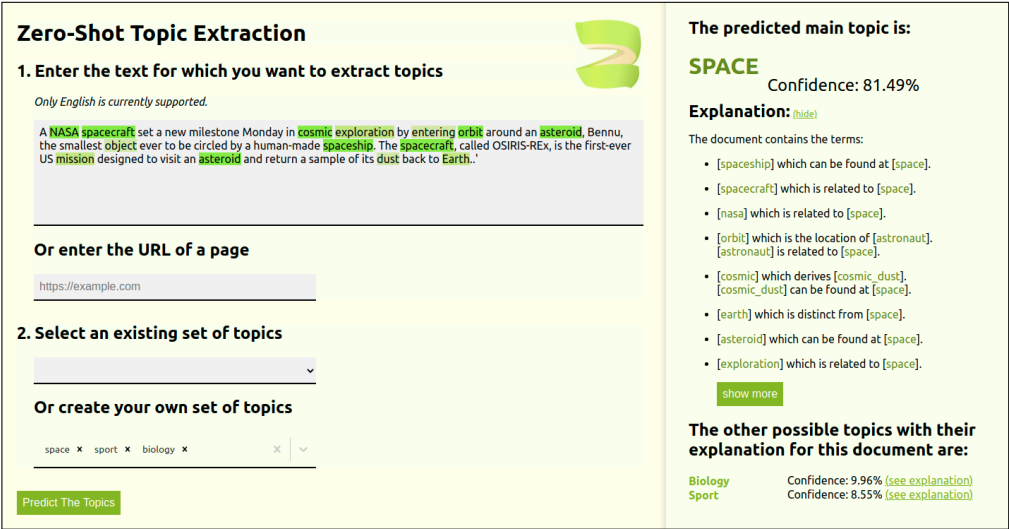
■ **Figure 3** The prediction error distribution along the normalized confidence scores

Dataset	BBC News	AG News	20 Newsgroups	AFP News
ZeSTE	80.6	71.0	61.6	73.8
ZeSTE + BERT	94.3	84.2	70.1	83.0

■ **Table 5** The accuracy of ZeSTE and used as bootstrapped model (using the generated predictions as training data) on the test split of each dataset

5 Online Demo

To demonstrate our method, we developed a web application which allows users to create their own topic classifier in real time. The user inputs the text to classify either by typing it into the designated textbox or by providing the URI of a web document that we scrape for extracting the content using Trafilatura⁷. The user is then prompted to either choose one of the pre-defined sets of labels (e.g. 20NG or IPTC used to evaluate the AFP dataset), or to provide her own set of label candidates. Once the user clicks on the "Predict the Topics" button, the server computes and caches the label neighborhood if it is the first time it encounters the label, otherwise it loads it from the cache for near real-time topic inference. Once the document is pre-processed and the label neighborhood generated, the server sends back its predictions (as confidence scores for each label candidate), and an explanation for each topic based on the common-sense connections between the document content and the label is provided (Figure 4, right panel). We only sample one path between document terms and the label, when in reality there could be many, in order to have a usable UI. In the future, we aim to depict the explanation as a subgraph of ConceptNet which shows all the relevant terms and their connections in the label neighborhood. We also highlight the relevant words in the input text (based on their score). While the demo works only for textual document written in English, we expect to support other languages in the future. The user interface makes use of the ZeSTE API which we also expose for others to be easily integrated.



■ **Figure 4** ZeSTE's User Interface deployed at <https://zeste.tools.eurecom.fr/>

⁷ <https://pypi.org/project/trafilatura/>

6 Conclusion and Future Work

In this work, we present ZeSTE, a novel method for zero-shot topic categorization that achieves competitive performance for this task, outperforming solid baselines and previous works while not requiring any labeled data. Our method also provides explainable predictions using the common-sense knowledge contained in ConceptNet. We demonstrate that ZeSTE can help to bootstrap a supervised classifier, achieving high accuracy on all datasets without requiring human supervision. The code to reproduce our approach and replicate our results is available at <https://github.com/D2KLab/ZeSTE>.

As an extension to this work, we consider an adaptation of the approach to other NLP tasks such as multi-class topic categorization, query expansion and keyphrase extraction. To further improve the approach, an analysis on how to partition the topic neighborhoods and minimise overlap is also envisaged. Finally, studying how to automatically pick better topic labels based on measures such as Mutual Information and Graph Centrality is to follow.

References

- 1 Katherine Bailey and Sunny Chopra. Few-shot text classification with pre-trained word embeddings and a human in the loop. arXiv:1804.02063, 2018. URL: <http://arxiv.org/abs/1804.02063>.
- 2 Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.
- 3 Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder. arXiv:1803.11175, 2018. URL: <http://arxiv.org/abs/1803.11175>.
- 4 Dawn Chen, Joshua C Peterson, and Thomas L Griffiths. Evaluating vector-space models of analogy. arXiv:1705.04416, 2017.
- 5 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186. Association for Computational Linguistics, 2019.
- 6 Zakaria Elberichi, Abdelattif Rahmoun, and Mohamed Amine Bentaalah. Using WordNet for Text Categorization. *International Arab Journal of Information Technology (IAJIT)*, 5(1), 2008.
- 7 Charles Elkan and Russell Greiner. Building large knowledge-based systems: Representation and inference in the Cyc project. *Artificial Intelligence*, 61(1):41–52, 1993.
- 8 Ingo Feinerer and Kurt Hornik. *wordnet: WordNet Interface*, 2017. R package version 0.1-14. URL: <https://CRAN.R-project.org/package=wordnet>.
- 9 Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *23rd International Conference on Machine learning (ICML)*, pages 377–384, 2006.
- 10 Antonio Gulli. *AG's corpus of news articles*, 2005. URL: http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.
- 11 Yoon Kim. Convolutional neural networks for sentence classification. arXiv:1408.5882, 2014.
- 12 Ken Lang. Newsweeder: Learning to filter netnews. In *12th International Conference on Machine Learning (ICML)*, pages 331–339, 1995.
- 13 Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On Measuring Social Biases in Sentence Encoders. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 622–628. Association for Computational Linguistics, 2019.

- 14 Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- 15 Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in Word Embeddings. In *International Conference on Fairness, Accountability and Transparency (FAT)*, pages 446–457. Association for Computing Machinery, 2020.
- 16 Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- 17 Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. Train once, test anywhere: Zero-shot learning for text classification. arXiv:1712.05972, 2017.
- 18 Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphaël Troncy, and Xavier Tannier. Searching News Articles Using an Event Knowledge Graph Leveraged by Wikidata. In *5th Wiki Workshop*, pages 1232–1239, 2019.
- 19 Vishal S Shirsat, Rajkumar S Jagdale, and Sachin N Deshmukh. Sentence level sentiment identification and calculation from news articles using machine learning techniques. In *Computing, Communication and Signal Processing*, pages 371–376. Springer, 2019.
- 20 Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pages 1223–1237, 2002.
- 21 Konstantinos Skianis, Fragkiskos Malliaros, and Michalis Vazirgiannis. Fusing document, collection and label graph-based representations with word embeddings for text classification. In *12th Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs)*, New Orleans, Louisiana, USA, 2018.
- 22 Yangqiu Song, Shyam Upadhyay, Haoruo Peng, Stephen Mayhew, and Dan Roth. Toward any-language zero-shot topic classification of textual documents. *Artificial Intelligence*, 274:133–150, 2019.
- 23 R. Speer and Joshua Chin. An Ensemble Method to Produce High-Quality Word Embeddings. arXiv:1604.01692, 2016.
- 24 Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *31st AAAI Conference on Artificial Intelligence*, 2017.
- 25 Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-Tune BERT for Text Classification? arXiv:1905.05583, 2019. URL: <http://arxiv.org/abs/1905.05583>.
- 26 Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online qa collections. In *46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 719–727, 2008.
- 27 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762, 2017. URL: <http://arxiv.org/abs/1706.03762>.
- 28 Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. arXiv:1902.07153, 2019.
- 29 Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.
- 30 Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. arXiv:1909.00161, 2019.
- 31 Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. arXiv:1903.12626, 2019.

B.10 EURECOM's SEMANTICS 2021 submission

This paper describes the exploratory search engine developed by EURECOM and submitted at SEMANTICS 2021.

KG Explorer: a Customisable Exploration Tool for Knowledge Graphs

Thibault Ehrhart¹, Pasquale Lisena¹[0000–0003–3094–5585], and Raphaël Troncy¹[0000–0003–0457–1436]

EURECOM, Sophia Antipolis, France
{ehrhart,lisena,troncy}@eurecom.fr

Abstract. The growing adoption of Knowledge Graphs demands new applications which enable users to search and browse structured data in a suitable way depending on the domain and application area. In this paper, we introduce KG Explorer, a web-based exploratory search engine for RDF-based Knowledge Graphs. The software can be configured in order to adapt to different information domains, customising both the UI components and the queries made for retrieving the information. The software also includes other features such as the ability to perform full-text search as well as facet-based advanced search in the data, and the possibility to create lists of favourites items modelled in the knowledge graph.

Keywords: knowledge graphs, data exploration, data access, search interface.

1 Introduction

Knowledge Graphs (KG) are more and more adopted for representing the information: today we can find several graphs, small and large, which may represent encyclopedic-general or domain-specific information. Their still growing popularity is due to an interesting set of characteristics, such as explicit semantic, interlinking with external resources, and a great expressiveness coming from Semantic Web technologies. KGs offer structured data that empower semantic search and QA systems among many other possible applications.

More recently, we see the interest in creating beautiful visualisation of KG-powered search results. An example is the use of Knowledge Panels on Search Engines, which are currently moving from simply displaying key-value tuples to integrate images and text for presenting the information nicely (Fig. 1).

Knowledge Graphs can be stored in dedicated triple store. Those, generally offer – next to the essential SPARQL endpoint – a browsing user interface (UI), which allows an end-user to see the loaded data on a web page. For example, the *facet browser* of Virtuoso¹ shows all incoming and outgoing predicates for a given resource with the respective values². When the value represents a picture,

¹ <http://vos.openlinksw.com/>

² Example from DBpedia: https://dbpedia.org/describe/?uri=http://dbpedia.org/resource/Antonio_Vivaldi



Fig. 1. The Knowledge Panel for the search keyword "Goat" in Google (left) and Bing (right), when configured with Country = USA and language = EN. Screenshot taken on 19/03/2021

the image is retrieved and displayed in the page. All entity nodes and edges are clickable, so that the user can navigate through the graph in a *follow-your-nose* approach. Other relevant features are plain text search (*/fct* on Virtuoso), a query helper (YASGUI³), dereferencing service for linked data URI, or rich visualization of results (like in Wikidata⁴).

However, these systems fall short when it is necessary to go beyond the simple visualisation of text and images and:

- embrace different media objects, such as video, audio, 3D graphics;
- propose new navigation paradigms, such as related items or recommendations for the next element;
- improve the search and exploration experience based on the domain peculiarities, filtering the results based on time ranges, geographic areas, or values from hierarchical thesauri. Moreover, the connection of the searched object and the value to filter can consist of a single direct property, a property path, or even a more complex query.

In order to provide a generic solution to these limitations, we introduce KG Explorer, a fully-customisable web application which serve as exploratory search

³ <https://triply.cc/docs/yasgui-api>

⁴ See for example <https://www.wikidata.org/wiki/Q2934>

engine [13] for Knowledge Graphs. KG Explorer offer alternative ways for navigating in a graph, enabling users to search and to follow links, to discover new information by exploiting the semantic proximity of entities.

The remaining of this paper is organised as follows. The related work is discussed in Section 2. We detail some requirements in Section 3. Section 4 describes the capabilities and functionalities of KG Explorer, while the architecture of the tool is explained in Section 5. A preliminary evaluation is carried on in Section 6. Finally, we conclude and outline some future work in Section 7.

2 Related Work

The VOILA workshop series⁵ has attracted a large number of specialized tools enabling to visualize linked data. An extensive survey of facet search has been published in [17]. This work has the merit of defining the basic concepts for the exploratory approach, namely the *extension* (the displayed results), the *intension* (the satisfied query) and the *transition markers*, clickable elements for triggering a transition (a new query). In addition, the work point out the possible kind of configuration of the tool, from the absence of any configuration requirement to the exact content to be displayed (view-based configuration).

Faceted Wikipedia Search [5] is a facet search tool based on DBpedia. The transition markers are sorted and displayed based on their frequency with respect to the number of results, in order to help the user in refining her/his query in successive iterations. Other provided features are free text search and range selection for datatype values. A similar interaction is implemented in *GraFa* [12], which refines the facet list after selecting the text keyword to search or the desired entity type. The involved schemas are indexed in order to have quick response, applying in addition a materialisation for the query returning the bigger number of results. These solutions are, however, based on statistics computed on properties, and do not take into account the domain specificity. In fact, the chosen facets are not always relevant nor useful for the search experience. The Metaphacts ecosystem⁶ includes an extension for building customisable apps on top of Linked Data.

FERASAT [7] shows the results obtained through combination of facet values in different visualisation components (maps, charts, etc.), in order to make evident the surprising results. This application targets a public of data experts but it would be quite complex for a broader audience. *LDVizWiz* [1] provides aggregate visualisations for entities of specific types in a KG, such as events which can be displayed on maps, timelines and tables. *Loupe* [11] displays the ontology classes and properties frequently used in tabular format, allowing the user to see how they are normally combined in the triples. These works show exclusively aggregate results, without enabling any customisation depending on the investigated domain.

⁵ <http://voila.visualdataweb.org/>

⁶ <https://metaphacts.com/>

In *Overture* [9], the visualisation of entity data is extended with custom components, showing a timeline of relevant events and the most similar entities from on a knowledge-based recommender system. In the *WarSampo portal* [8] (about Finnish history in World War II), different tabs allows to switch between a tabular visualisation of data, a timeline and a map, and a photo gallery⁷. The resource page of *Genesis* [4] includes entity textual data, images and videos, as well as a selection of similar and related entities with their own depictions. These examples are ad-hoc developed tools, hard to adapt to new domains.

The Fresnel vocabulary [14] has been proposed for closing the gap between data and presentation, enabling to define content subsets and formats matched with CSS classes. Similarly, custom views are used for driving the visualisation in [2,16,3]. However, these approaches do not propose solutions to data search.

3 Different Scenarios But Shared Needs

Different users may take advantage from data inside specialised Knowledge Graphs, each one with their own needs and goals. We identified the following shared needs:

- to understand what is in the dataset, and in particular the main resource types (classes) and how they are connected to each other;
- to search for specific resource which satisfy some domain-relevant criteria;
- to obtain detailed information about a particular resource, including multimedia data and smart aggregations using timelines, maps and plots.

These need are highly impacted by the kind of user, which can fall in one of the following scenarios:

- *domain experts* have great interest in the subject, are used to the domain vocabulary and know what they search with precision. They need advanced search capabilities, allowing them to filter the results by several dimensions. The information needs to be complete.
- the *wide public* is rather moved by the curiosity of discovering something new, sometimes having only general or null knowledge about the domain. They need to easily browse the data collection and possibly reach relevant information already after the first click. Some strategies are needed to make them continue the exploration, for example follow-your-nose approaches or the recommendation of similar or related items. The engagement is crucial for their experience.
- *external stakeholders* need to know which relevant information is possible to find in the data and how to easily access it.

We argue that an exploratory search engine [13] enables to fulfil the described needs while being flexible enough to targeting the different personas. In addition, the application should have a proper user interface (UI), which reflect the domain

⁷ Example: https://www.sotasampo.fi/en/persons/person_61

specificity and the institution identity. In the same time, this can improve the final user engagement. Further requirements are the selection of the language for KGs including multi-lingual contents and an authentication method for data that are not public.

4 KG Explorer Functionalities

Having defined the expected requirements, we are going to detail in this section the features implemented in KG Explorer: a facet-based advanced search engine, dedicated editorial pages for controlled vocabularies represented in SKOS and generally used in the knowledge graph, a customised detailed page for the main entities represented in the knowledge graph, the possibility for users to log in and to create personalized lists of favourites or saved items. The software can be configured to adapt to different information domains, changing not only its aspect but also the queries for retrieving the data to display. KG Explorer is open source under Apache License 2.0 at <https://github.com/D2KLab/explorer>. In order to explain the software capabilities, we will refer to three in-use applications of KG Explorer. These examples use data coming from different domains (cultural heritage, television and news), each of them with proper customisation. The links to the applications and the source code are collected in Table 1.

ontologies	#entities	links
ADASilk - <i>domain</i> : silk heritage		
CIDOC-CRM CRMsci	675,112	Source code: https://git.io/adasilk Application: https://ada.silknow.org/
MeMAD Explorer - <i>domain</i> : TV and Radio programmes		
EBUcore	1,079,969	Source code: https://git.io/memad-explorer Application: https://explorer.memad.eu/
ASRAEL Search Engine - <i>domain</i> : news and events		
OpenAnnotation rNews schema.org	968,602	Source code: http://bit.ly/asrael-se Application: http://asrael.eurecom.fr/search-engine

Table 1. In-use instances of KG Explorer (including ASRAEL Search Engine which is a fork of the main tool).

4.1 A standardised experience

KG Explorer offers a user experience based of four different kind of pages. The **landing page** contains a search box which allows the user to perform a free text search on entities modeled in the Knowledge Graph. When the user enters a search term, the exploratory search engine executes a SPARQL query with a REGEX filter in order to select all items that have a label or a title that partially

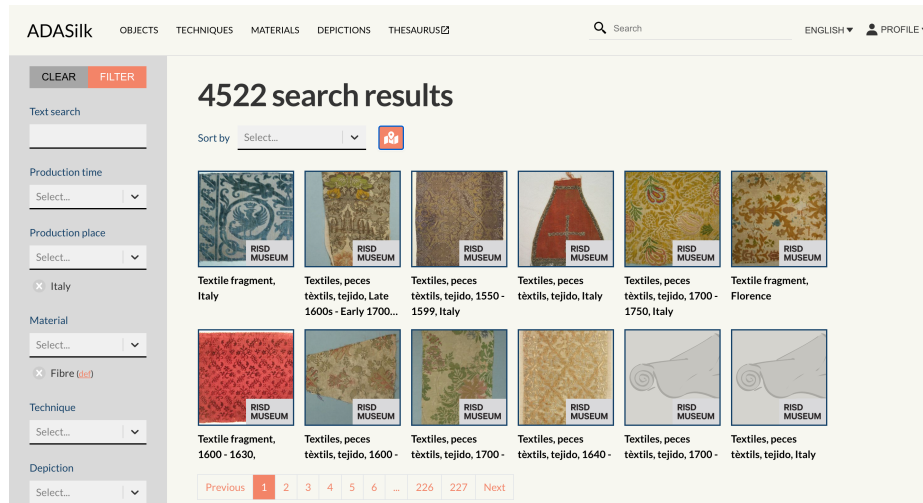


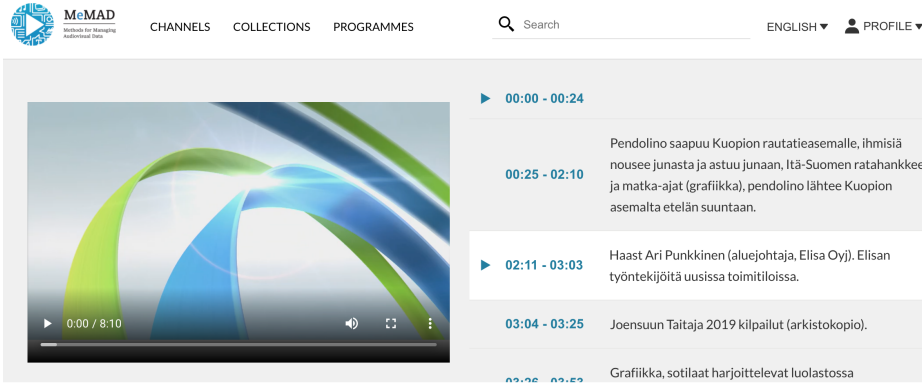
Fig. 2. The browse page in ADASilk.

matches the search terms. The search query algorithm can also be changed in the configuration file to cover all datatype properties of the graph. The results are shown in an auto-complete box.

The **browse page** (Fig. 2) contains a faceted search engine which allows users to perform an advanced search for the main entities of the Knowledge Graph. The sidebar on the left side contains facets (or filters). Each facet generates an extra condition to the main SPARQL query used for searching.

In addition to a textual search box, the exploratory search engine provides shortcuts to so-called **vocabulary pages**, which show all terms belonging to a particular thesaurus – e.g. a *ConceptScheme* in the SKOS namespace. These vocabularies are defined in the configuration file, and are usually materialised as concepts in the Knowledge Graphs. Clicking on a vocabulary term will typically bring the user to a pre-filtered browse page, in order to see the related items in the knowledge graph.

Finally, the **detail page** shows all the information related to a single entity. There are currently 3 layouts available for detail pages: collection (grid-based list of items), gallery (carousel of images), and video (media player, as in Fig. 3). Custom pages can be added by creating new JavaScript files in the `pages/` directory, and exporting the class as a React component. Each page is automatically included in the build and associated with a route based on its file name. New layouts can also be added to the project, by creating a new file in the `pages/details/` directory, and referring to its name in the `view` property in the configuration file. This is being used for developing the video player view in the MeMAD Explorer, handling also authentication to the media server.



Yle Uutiset Itä-Suomi, Yle Nyheter östra Finland

[\(permalink\)](#)

EPISODE NUMBER
100

DESCRIPTION
Uutisia Itä-Suomesta

Related

Yle Uutiset Itä-Suomi, Yle Nyheter östra Finland

00:00 - 00:24

00:25 - 02:10

02:11 - 03:03

03:04 - 03:25

03:26 - 03:59

Pendolino saapuu Kuopion rautatieasemalle, ihmisiä nousee junasta ja astuu junaan, Itä-Suomen ratahankkeen ja matka-ajat (grafiikka), pendolino lähtee Kuopion asemalta etelään suuntaan.

Haast Ari Punkkinen (aluejohtaja, Elisa Oyj). Elisan työntekijöitä uusissa toimitiloissa.

Joensuun Taitaja 2019 kilpailut (arkistokopio).

Grafiikka, sotilaat harjoittelevat luolastossa

Fig. 3. The detail page in MeMAD Explorer, with the video player view.

4.2 User profiles

KG Explorer includes an authentication system which allows users to create an account, log in and have access to additional features. The OAuth authentication method is used for creating a new profile and for any successive login, relying on signing-in via Google, Facebook, and Twitter. Once logged in, users have the possibility to create named lists for storing searched items. A "save" button is present on each detail page, allowing to add the current page to an existing list or to create a new one. Lists can be retrieved in the profile page of the user, from where they can also be made public and shared with anyone using a permalink. Moreover, from the profile page it is possible to link or unlink additional OAuth accounts, as well as manage the existing lists or even delete entirely the user profile.

4.3 Generic tool, custom configuration

Each domain and KG has its own characteristic. KG explorer is capable of working on top of any RDF-based Knowledge Graph, by configuring an instance of it using a JavaScript file (`config.js`). The configuration allows to define a wide set of options, such as the chosen SPARQL endpoint, the supported language for internationalisation, and some layout-related settings – i.e. which images to use, which components to show or hide, etc.

```

{
  objects: {
    view: 'browse', // type of view ('browse' or 'vocabulary')
    showInNavbar: true,
    rdfType: 'http://erlangen-crm.org/current/E22_Man-Made_Object',
    uriBase: 'http://data.silknow.org/object',
    details: { view: 'gallery' },
    filters: [{ // set of filters to appear in the advanced search
      id: 'material', // material filter
      isMulti: true, // 1 or more values can be selected
      isSortable: true,
      vocabulary: 'material', // values taken from a vocabulary
      whereFunc: () => [ // added to the base query when filtering
        '?production ecrm:P126_employed ?material',
        `OPTIONAL {
          ?broaderMat (skos:member|skos:narrower)* ?material }`
      ],
      filterFunc: (values) => { // add to base query when filtering
        return [values.map((val) =>
          `?material = <${val}> || ?broaderMaterial = <${val}>`)
          .join(' || ')];
      }
    ]],
    baseWhere: [
      'GRAPH ?g { ?id a ecrm:E22_Man-Made_Object }',
      '?production ecrm:P108_has_produced ?id',
    ],
    query: { // base query
      '@graph': [{
        '@type': 'http://erlangen-crm.org/.../E22_Man-Made_Object',
        '@id': '?id',
        '@graph': '?g',
        label: '$rdfs:label',
        identifier: '$dc:identifier',
        description: '$ecrm:P3_has_note',
      }],
      $where: ['GRAPH ?g { ?id a ecrm:E22_Man-Made_Object }']
    }
  }
}

```

Listing 1: Partial definition of the ‘Objects’ route in ADASilk, with the optional filter by material

Of particular interest is the possibility of defining the pages that compose application, through the `route` field of the configuration file. The example in Listing 1⁸ shows the available options, which include the choice between *browse* or *vocabulary* page, the page URI, the applied JSON query for listing the results (following the SPARQL Transformer syntax, as described in Section 5).

In *browse* pages, the `filters` property can contain a list of available fields for the advanced search, detailing also which changes are applied to the query when filters are applied. The list of available values can be loaded with a query (defined or made globally available as *vocabulary*). The main query condition is defined with the *baseWhere* property, with the minimal amount of triples required in order to improve performances. Once the list of results has been fetched, a second query is made to get the details of each result. This query is defined within the *query* property. The labels for the **internationalisation** are collected in specific JSON files to include in the project directory.

The front-end also supports **custom styles** which can be defined in a `theme.js` file. This allows to further customise the appearance of the user interface. It is possible to choose the global font set and a custom colour palette. Moreover, specific components can also be customised, by using the name of component and defining CSS rules following the *styled-components* syntax. Finally, adding custom pages and view (Section 4.1) enable the developer to include new **visualisation components**. Examples are maps and 3D visualisation in ADASilk.

5 Architecture

Fig. 4 shows an overview of the architecture and the technologies used in KG Explorer. KG Explorer is developed in a **containerised approach**, implemented within the Docker framework⁹: thanks to the use of independent and self-sufficient containers, Docker enables the deployment of this architecture on any machine, automatically installing and running the required software. This approach also allows to easily extend and deploy new instances of the application from the base image, including custom configuration and assets, as has been done in the instances in Table 1.

The **web application** is composed of several web technologies. The front-end is produced using *React*, a JavaScript library for building user interfaces¹⁰. It uses encapsulated components that manage their own state to help maximise code re-usability. The framework *Next.js*¹¹ is used for server-side rendering and page-based routing. It relies on a file-based structure for pages routing, where each page has its own file which is stored in the `src/pages` directory. Additionally, special routes are dedicated to serve APIs, used on the server-side for

⁸ The code is extracted from the ADASilk configuration and is fully available at <https://github.com/silknow/adasilk/blob/main/config/routes/object.js>

⁹ <https://www.docker.com/>

¹⁰ <https://reactjs.org/>

¹¹ <https://nextjs.org/>

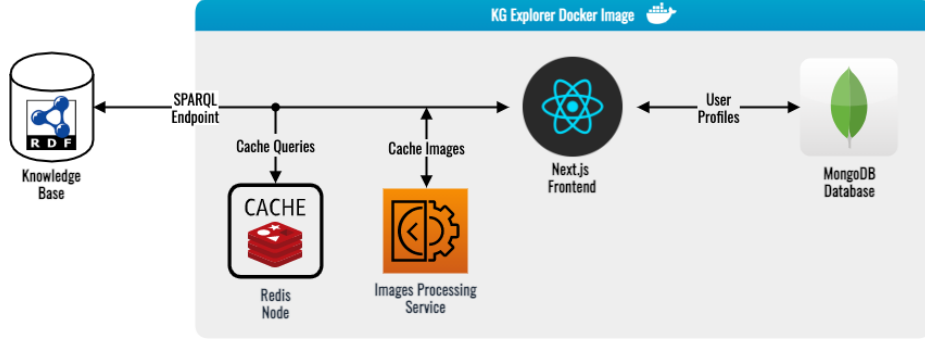


Fig. 4. Architecture of KG Explorer

handling authentication, fetching profile data, and searching for items. The library *styled-components*¹² is used for styling React components using scoped CSS (Cascading Style Sheet). It relies on tagged template literals to inject CSS code inside pre-defined components. Other used frameworks are *i18next*¹³ for the internationalisation and *next-auth*¹⁴ for OAuth authentication.

KG Explorer makes requests to a Knowledge Graph through its exposed SPARQL endpoint. In order to easily include and manipulate queries in JavaScript, those are written in the JSON query syntax proposed by **SPARQL Transformer** [10]. The SPARQL Transformer library makes it easy to define queries using JavaScript objects (called *JSON queries*) which can be edited and merged to create the final query. For instance, each filter from the faceted search appends its own conditions to the base query, as seen in Section 4.3. Looking again at Listing 1, when a filter is applied, the base query is modified applying new **WHERE** and **FILTER** expressions, respectively defined in **whereFunc** and **filterFunc**. The use of JSON queries makes it possible to simply append this expression in the **\$where** and **\$filter** properties of SPARQL Transformer, and avoids a much more complex manipulation of text which the use of plain SPARQL queries would require. SPARQL Transformer also rewrites the output of SPARQL queries in a more suitable format for web development. In particular, SPARQL results composed of bindings between variables and solutions are transformed into self-contained JSON objects, including all the information about the entities, getting rid of some verbosity of the standard notation. Queries results are processed and cached into a *Redis* database¹⁵ in order to improve performances. The results are stored as a JSON string, and the original query is used as the key for retrieving the cached result. User profiles and lists are instead saved in a *MongoDB* document-based database¹⁶.

¹² <https://styled-components.com/>

¹³ <https://www.i18next.com/>

¹⁴ <https://next-auth.js.org/>

¹⁵ <https://redis.io/>

¹⁶ <https://www.mongodb.com/>

6 Preliminary Evaluation

Preliminary evaluations of KG Explorer were conducted as part of the SIL-KNOW project and reported in [15]. The application has been used by 216 users, reflecting different audience, domain and technical skills (Table 2). The users were asked to perform a number of search activities and to comment on the results reflecting both the intrinsic quality of the knowledge graph which is hard to isolated and the ability of searching for specific items and of browsing and discovering new items.

Domain	English	French	Spanish	Italian	Total
Cultural Heritage	0	0	14	14	28
Education related to social science	1	0	6	4	10
Information and communication technology	1	17	42	67	126
Textile or creative industry	0	1	1	1	3
Tourism	0	1	0	2	3
Media	0	2	2	3	7
Other	0	2	15	22	39
					216

Table 2. Target audience used during the evaluation of ADASilk.

During the evaluation phase, each user session has also been recorded, after consent, so that it could be analyzed later. To do this, the `rrweb`¹⁷ library is implemented into the UI in order to record and then replay each interaction with the interface. The recorded sessions are saved as JSON objects in a database. At the end of the evaluation, the sessions were exported as MP4 videos using `rrvideo`.¹⁸ We report below the most common issues and what users perceive as anomalous behaviour.

From the analysis of all the tests conducted through ADASilk, a commonly encountered issue is related to the text search functionality. While offering free text search was found to be an essential feature, it also raises some expectations that the search query will be somehow interpreted. Users are familiar with Google which interprets and disambiguates search queries while offering personalized answers. In contrast, KG Explorer offers either a naive *text search* that aims to match resources for which the search terms can be encountered in a datatype property value or a *concept search* which can lookup and auto-complete concepts from controlled vocabularies typically used in facets. Often, users have entered simple search strings expecting that their translations in other languages will bring the same result set.

The relevance of the search results was also pointed out as an issue during the evaluation, in particular, by domain experts. The sole SPARQL query language

¹⁷ <https://github.com/rrweb-io/rrweb>

¹⁸ <https://github.com/rrweb-io/rrvideo>

offers only the possibility of returning a set of exact solutions to a query without natural ways of ranking the resources within this set nor with the possibility to consider partially related resources. The numerous methods enabling to build knowledge graph embeddings are promising to bring this notion of relevance, e.g., in measuring the distance between each document. We observe that some triple stores, such as GraphDB¹⁹, have started to provide native support for semantic similarity searches.

7 Conclusion and Future Work

KG Explorer provides a domain-specific user experience for exploring the information contained in a Knowledge Graph. The software can be easily customised and adapted in the UI and in the content, defining the queries for retrieving the data, the facets to be used, and the relevant vocabularies. KG Explorer is already used in real-world applications, in particular as wide-public entry-point for Knowledge Graphs of research projects. In this context, a user evaluation is currently being carried out where the goal is to measure the usability of the application in the fulfilment of common tasks, identified by domain experts. The outcome of this evaluation will be used for further improving the application.

Future developments will also involve new functionalities such as having custom facet selectors for datatypes, for example ranges for numbers and dates. Finally, we would like to exploit the vocabularies in order to provide a smart text search field, going beyond the simple exact match on text: this can be implemented by recognising terms defined in vocabularies and attaching them to the most appropriate property in the generated query, in a query interpretation behaviour. In this field, previous research has proved the suitability of embedding techniques for representing a query, in order to get more relevant results [6,18].

Acknowledgements

This work has been partially supported by the European Union’s Horizon 2020 research and innovation program within the SILKNOW (grant agreement No. 769504) and MeMAD (grant agreement No. 780069) projects, and by the French National Research Agency (ANR) within the ASRAEL project (grant number ANR-15-CE23-0018).

References

1. Atemezing, G.A., Troncy, R.: Towards a Linked-Data Based Visualization Wizard. In: 5th International Conference on Consuming Linked Data (COLD). Riva del Garda, Italy (2014)

¹⁹ <https://graphdb.ontotext.com/>

2. Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D.: Tabulator: Exploring and Analyzing linked data on the Semantic Web. In: 3rd International Semantic Web User Interaction Workshop (SWUI) (2006)
3. Chauvat, N., Amarger, F., Wouters, L.: Un navigateur pour le Web des données liées. In: 30es Journées Francophones d'Ingénierie des Connaissances, IC 2019. pp. 167–182. Toulouse, France (2019)
4. Ermilov, T., Moussallem, D., Usbeck, R., Ngonga Ngomo, A.C.: GENESIS: A Generic RDF Data Access Interface. In: International Conference on Web Intelligence (WI). pp. 125–131 (2017)
5. Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted Wikipedia Search. In: 13th Conference on Business Information Systems (BIS) (2010)
6. Hamilton, W.L., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding Logical Queries on Knowledge Graphs. In: 32nd International Conference on Neural Information Processing Systems (NIPS). pp. 2030–2041 (2018)
7. Khalili, A., van den Besselaar, P., de Graaf, K.A.: FERASAT: A Serendipity-Fostering Faceted Browser for Linked Data. In: 17th International Semantic Web Conference (ISWC). pp. 351–366 (2018)
8. Koho, M., Ikkala, E., Leskinen, P., Tamper, M., Tuominen, J., Hyvönen, E.: WarSampo knowledge graph: Finland in the Second World War as Linked Open Data. *Semantic Web Journal* pp. 1–14 (2020)
9. Lisena, P., Achichi, M., Fernandez, E., Todorov, K., Troncy, R.: Exploring Linked Classical Music Catalogs with OVERTURE. In: 15th International Semantic Web Conference (ISWC), Posters & Demos Track. Kobe, Japan (2016)
10. Lisena, P., Meroño-Peñuela, A., Kuhn, T., Troncy, R.: Easy Web API Development with SPARQL Transformer. In: 18th International Semantic Web Conference (ISWC). pp. 454–470. Auckland, New Zealand (2019)
11. Mihindukulasooriya, N., Poveda-Villalón, M., García-Castro, R., Gómez-Pérez, A.: Loupe - An Online Tool for Inspecting Datasets in the Linked Data Cloud. In: 14th International Semantic Web Conference (Posters & Demos) (2015)
12. Moreno-Vega, J., Hogan, A.: Grafa: Scalable faceted browsing for rdf graphs. In: 17th International Semantic Web Conference (ISWC). pp. 301–317 (2018)
13. Palagi, E., Gandon, F., Giboin, A., Troncy, R.: A Survey of Definitions and Models of Exploratory Search. In: ACM Workshop on Exploratory Search and Interactive Data Analytics (ESIDA). Limassol, Cyprus (2017)
14. Pietriga, E., Bizer, C., Karger, D., Lee, R.: Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In: 5th International Semantic Web Conference (ISWC). pp. 158–171 (2006)
15. Seidita, V., Lo Cicero, G., Vitella, M.: Testing Report in a Real Scenario. project deliverable D7.2, H2020 SILKNOW (2021)
16. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: Live views on the Web of Data. *Journal of Web Semantics* 8(4), 355–364 (2010)
17. Tzitzikas, Y., Manolis, N., Papadakis, P.: Faceted exploration of rdf/s datasets: A survey. *Journal of Intelligent Information Systems* 48(2), 329–364 (2017)
18. Xiong, C., Power, R., Callan, J.: Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In: 26th International Conference on World Wide Web (WWW). pp. 1271–1279. Perth, Australia (2017)

B.11 Aalto's TSD 2021 submission

This paper describes the Aalto's spoken NER models submitted to TSD 2021.

Attention-Based End-To-End Named Entity Recognition From Speech

Firstname1 Surname1, Firstname2 Surname2, and Firstname3 Surname3

Affiliation1, Institute1, Address
www.website.org
{author1, author2}@institutel.org

Abstract. Named entities are heavily used in the field of spoken language understanding, which uses speech as an input. The standard way of doing named entity recognition from speech involves a pipeline of two systems, where first the automatic speech recognition system generates the transcripts, and then the named entity recognition system produces the named entity tags from the transcripts. In such cases, automatic speech recognition and named entity recognition systems are trained independently, resulting in the automatic speech recognition branch not being optimized for named entity recognition and vice versa. In this paper, we propose two attention-based approaches for extracting named entities from speech in an end-to-end manner, that show promising results. We compare both attention-based approaches on Finnish, Swedish, and English data sets, underlining their strengths and weaknesses.

Keywords: Named entity recognition, Automatic speech recognition, End-to-end, Encoder-decoder

1 Introduction

Named entity recognition (NER) is one of the main natural language processing (NLP) tasks. The goal of this task is to find entities and classify them into predefined categories. These categories can vary depending on the application area, but the most common ones include person, location, organization, and date.

Named entities are heavily used in spoken language understanding (SLU) [3] [15] [9], where the goal is to understand what has been spoken. For example, SLU is an essential part of personal assistants in home automation and smartphone devices. These personal assistants usually take speech as input, in which case the named entities need to be recognized from spoken data.

Doing NER from speech imposes several challenges for the system. There are far fewer annotated training data for spoken language than for textual data. The speech can be informal, not following the conventional syntax of the language, which can cause difficulties in detecting the entities. The generated transcripts from an automatic speech recognition (ASR) system usually do not contain capitalization and punctuation, which can cause the system to miss the entities.

The most common approach for doing named entity recognition from speech is through a pipeline approach. In this approach, the ASR system generates transcripts,

and the NER system detects the entities in those transcripts. The output of the ASR system is usually lower-cased and noisy, in the sense that the word order can be mixed, words might be missing or misspelled, etc. When developing a NER system for speech data, these factors need to be taken into account.

It is possible to try to restore the capitalization and the punctuation from the transcribed speech as explored in [6]. A maximum entropy model was used for NER on transcripts generated by a speech recognition system for Chinese, utilizing n-best lists [22]. These approaches improve the performance of the system on noisy speech data but they are still sensitive to the speech recognition output and error propagation. To deal with that, an end-to-end (E2E) approach was proposed that directly extracts named entities from French speech [5]. The authors used an architecture similar to the Deep Speech 2 [1], which was trained using the CTC algorithm [7]. A similar approach of E2E named entity recognition using the Deep Speech 2 architecture for the English language was explored in [21]. This is different from our proposed models, which use either attention-based encoder-decoder (AED) or a hybrid CTC/AED architecture.

In this paper, we propose two approaches for doing E2E NER from speech. To the best of our knowledge, this is the first attempt at NER using AED architecture in an E2E manner. The first approach is called augmented labels (AL) and it is either a standard AED or a hybrid CTC/AED architecture, where the transcripts are augmented with named entity tags during training. The second is a multi-task (MT) approach, where there are two decoder branches. One branch for doing automatic speech recognition and another one for doing named entity recognition.

2 Data

In our experiments, we used four data sets for three different languages: Finnish, Swedish, and English.

For the Finnish experiments, we used the Finnish parliament data set [14], consisting of about 1500 hours of recordings from the Finnish parliament. Since we do not have true named entity labels for this data set, we used a separate NER system to annotate it. The NER system is a bidirectional LSTM (BLSTM) neural network [8] with a Conditional random field (CRF) [11] layer on top, that utilizes morph, character and word embeddings. The architecture is explained in more detail in [17]. The number of tokens and named entity tags in the data set are presented in Table 1.

Table 1. Data distribution for the Finnish parliament data set.

Parameters	Count
Audio length	1500 h
Total tokens	7.3 M
Unique tokens	337423
PER tags	44984
LOC tags	73860
ORG tags	65463

For the Swedish experiments, we used the Sprakbanken corpus, which is a public domain corpus hosted by the National Library of Norway. It consists of 259 hours of recordings. Since the corpus does not contain ground truth named entities, we used the Swedish BERT model [13] to obtain the annotations. The number of tokens and named entity tags are presented in Table 2.

Table 2. Data distribution for the Swedish data set.

Parameters	Count
Audio length	259 h
Total tokens	1.4 M
Unique tokens	69310
PER tags	23258
LOC tags	7585
ORG tags	2231

Even though the goal of this paper is mainly focused on low-resource languages like Finnish and Swedish, we additionally wanted to verify the performance of the models on a well-known language, like English.

For the English experiments, we used the whole LibriSpeech data set [16], consisting of about 1000 hours of recordings. The named entities for this data set were obtained using the large uncased BERT model [4], fine-tuned on the CoNLL 2003 data set [18], which we lower-cased before training. For testing the model with gold-standard named entity tags, we used a data set which is a subset of a combination of multiple speech recognition data sets, such as CommonVoice, LibriSpeech, and Voxforge. We will call this data set English-Gold. The data set is annotated and provided by [21]. The number of tokens and named entity tags in the English data sets are presented in Table 3.

Table 3. Data distribution for the English LibriSpeech and English-Gold data sets.

Parameters	LibriSpeech	English-Gold
Audio length	1000 h	148 h
Total tokens	9.6 M	1.3 M
Unique tokens	87600	41379
PER tags	194172	50552
LOC tags	66618	23976
ORG tags	11415	5025

3 Methods

To do E2E named entity recognition from spoken data, we will explore two approaches. In the first approach, we will build an attention-based encoder-decoder model for ASR

by augmenting the labels with NER tags. In the second approach, we will explore multi-task learning where the model simultaneously learns to transcribe speech and annotate it with named entity tags. Additionally, for the English and Swedish experiments, we utilize the CTC loss, as explored in [20].

Generally, the E2E ASR models can benefit from an external language model [19] but in our experiments we exclude it. The reason for that is because the augmented labels approach produces an output where each word is followed by a named entity tag. In such a case, adding an external language model trained on text will not benefit us. On the other hand, the baseline ASR models can benefit from an external language model but the goal of this paper is to explore an alternative way of doing named entity recognition from speech, as opposed to the standard pipeline approach.

3.1 Pipeline NER Systems

To see how our proposed models perform in comparison to the pipeline approach, where an ASR system generates the transcripts and then a NER system annotates them, we trained BLSTM-CRF models for each of the data sets. The architecture of these models is identical to the NER branch in the multi-task approach, described later in the paper. The models are trained on the original transcripts for each of the data sets. Since the English-Gold data set is small, we used the LibriSpeech model to initialize the weights and then fine-tune it on that particular data.

3.2 Baseline ASR System

The baseline ASR architecture is the same as the augmented labels approach, which is explained later in the paper. The only difference is that for the training of the baseline models, we used the original transcripts, whereas for the augmented labels approach we used the original transcripts augmented with named entity tags. We choose the architectures to be identical so that we can give a fair comparison between them.

3.3 Augmented Labels Approach

For this approach, we developed an attention-based encoder-decoder architecture that takes audio features as input and produces transcripts with named entity tags. Let $X = (x_1, x_2, \dots, x_T)$ be the audio features, where each feature is represented as x_i and i is the order of the feature. Additionally, we define the output character set $Y = (y_1, y_2, \dots, y_T)$, where y consists of all the characters plus the special tokens: $\langle \text{UNK} \rangle$, $\langle \text{eos} \rangle$, $\langle \text{eos} \rangle$, O, PER, LOC, and ORG. The goal is to model the conditional probability:

$$P(Y|X) = \prod_i P(y_i | Y_{<i}, X) \quad (1)$$

In simpler terms, it predicts the i -th output character, given the previous characters and the input features X . It does this using an encoder and a decoder.

The encoder is a BLSTM neural network, that uses audio features as input and compresses them in a single hidden representation. This hidden representation is used to initialize the decoder.

The decoder is an LSTM neural network that takes the hidden vector, produced by the encoder and generates the transcripts using an attention mechanism. As an attention mechanism, we used Luong attention [12]. The scoring function for the attention is hybrid + location-aware, as described in [2]. It is defined as:

$$score(h_{enc}, h_{dec}) = v * tanh(W^e * h_{enc} + W^d * h_{dec} + W^c * conv + b) \quad (2)$$

where, h_{enc} and h_{dec} are the hidden states of the encoder and the decoder, $tanh$ is a hyperbolic tangent non-linearity, v and b are learnable weights, together with the W matrices. The location-aware element $conv$ is a convolution defined as:

$$conv = F * \alpha_t \quad (3)$$

where, F is a learnable matrix and α_t is the alignment vector.

For the experiments where we additionally used the CTC loss, the final ASR loss is calculated as:

$$L_{asr} = \lambda L_{ctc} + (1 - \lambda) L_{aed} \quad (4)$$

where, L_{ctc} is the CTC loss, L_{aed} is the decoder loss and λ is the weighting factor that determines the contribution of the separate loss functions to the final loss.

As true labels, we used the transcripts, augmented with named entity tags, in a way that each word is followed by its tag. This way, the model will jointly produce ASR transcripts and NER tags.

3.4 Multi-Task Approach

The multi-task approach is an attention-based encoder-decoder architecture, similar to the augmented labels approach. The difference between them is that this approach has two separate decoder branches. The first branch does the automatic speech recognition and is like the one in the augmented labels. The second one does the named entity tagging and it consists of BLSTM with a CRF layer on top. This approach uses hard parameter sharing, where the encoder is shared between both branches. Since it is a multi-task learning approach, we have two separate loss functions that need to be jointly optimized. The final loss function is calculated as:

$$L = \beta L_{asr} + (1 - \beta) L_{ner} \quad (5)$$

where L_{asr} is the loss from the ASR decoder, L_{ner} is the loss from the NER decoder, and β is a weighting factor that determines the contribution of both loss functions.

Similar to the augmented labels approach, in the experiments where we utilized the CTC loss, the ASR loss L_{asr} is calculated as in Equation 4.

4 Experiments

In all the experiments, we used logarithmic filter banks with 40 filters and Adam optimizer [10]. For the multi-task approach, after the models converged, we additionally froze the encoder and the ASR decoder and trained only the NER branch, which improved the multi-task NER results on most of the data sets. We will refer to this model as MT*. The code was developed using Pytorch and is publicly available.¹

Speech features consist of a large number of timesteps, so processing them using a standard BLSTM network is computationally expensive. To deal with that we used a pyramidal BLSTM network. The pyramidal structure reduces the computational time by concatenating every two consecutive timesteps in each layer.

In the Finnish and English experiments, the encoder consists of 5 pyramidal BLSTM layers, whereas in the Swedish experiments we used 3 normal and 2 pyramidal BLSTM layers. The reason for that is because the Swedish data set consists of short utterances, so there are not many timesteps to be processed. The hidden size of the BLSTM networks is 450 in all the experiments, except for the Finnish, where we used a hidden size of 300. After the last BLSTM layer, a dropout of 0.1 is applied.

In the augmented labels approach, the decoder consists of a character embedding layer with a size of 150 and a single layer LSTM network. For the English and Swedish experiments, the LSTM has a size of 450, whereas for the Finnish experiments, it has a size of 300. The location-aware element in the attention has 150 filters for the English and Swedish, and 100 filters for the Finnish experiments. A dropout of 0.1 is applied after the attention mechanism.

In the multi-task approach, the ASR decoder is identical to the one in the augmented labels, for all the experiments. The NER decoder uses pre-trained 300 dimensional fastText word embeddings as an input to the one-layer BLSTM. The size of the BLSTM layer is 450 for the English and Swedish experiments, and 300 for the Finnish ones. The BLSTM is followed by a fully connected layer with the same size and a dropout layer with a probability of 0.1. In the end, the output is passed through a CRF layer that produces the tag probabilities.

Since the English-Gold data is relatively small with only 148 hours, we used the LibriSpeech data to pre-train the model and then fine-tune it on the English-Gold data set.

In all the experiments, we allocated data for testing, which was not used during training. As a loss function, we used the negative log-likelihood. For combining the ASR and NER losses, as in Equation 5, we used β weighting factor of 0.8. For the Swedish and English experiments, we additionally utilized the using CTC, together with negative log-likelihood, like in the Equation 4, with a λ weighting factor of 0.2.

5 Results

In this section we present the results obtained on Finnish, Swedish, and English data sets, comparing both the augmented labels and multi-task approaches. For the evalua-

¹ XXX

tion of the ASR results, we used the word error rate (WER) metric, and for the evaluation of the named entity recognition results, we used the micro average F1 score.

5.1 Finnish Results

In Table 4, we can see how both the augmented labels and multi-task approaches compare against the baseline ASR model in terms of WER when evaluated on the Finnish parliament data. From the results, we can notice that both approaches perform in pair with the baseline ASR model, falling slightly behind. We can also see that the multi-task approach performs slightly better than the augmented labels approach in terms of WER.

In Table 5, we can see how both approaches perform in terms of precision, recall, and F1 score. Additionally, we evaluated our models on the original transcripts and on the transcripts that were generated by the models. We used the multi-task and the fine-tuned multi-task models to do the evaluation on the original transcripts. From the results, we can see that the fine-tuned multi-task model performs slightly better than the standard multi-task model. On the transcripts generated by the model, which is a harder task, we compared both multi-task approaches, along with the augmented labels and the pipeline approach. The ASR transcripts for the pipeline approach were generated using the multi-task model, for all the data sets. From the results, we can see that the fine-tuned multi-task approach achieved the best F1 score. We can also notice that both multi-task approaches perform better than the pipeline approach, whereas the augmented labels approach falls behind.

Table 4. WER on the Finnish test set.

Model	WER
Baseline ASR	34.95
AL	36.06
MT	35.80

Table 5. Precision, recall and F1 score for the Finnish test set.

Transcripts	Model	Prec	Rec	F1
Original	MT	93.70	92.88	93.29
	MT*	93.75	93.69	93.72
Generated	Pipeline	93.63	85.64	89.46
	AL	92.65	81.61	86.78
	MT	93.35	87.80	90.49
	MT*	93.17	88.80	90.93

5.2 Swedish Results

Next, we present the Swedish results. In Table 6, we can see how both approaches perform in terms of WER, in comparison to the baseline model. Similar to the Finnish experiments, we can see that both models fall slightly behind the baseline ASR model. Additionally, we can observe that the augmented labels approach performs better than the multi-task approach.

From Table 7, we can see how our models perform on the NER task when evaluated on the original and the generated transcripts. When evaluated on the original transcripts, the fine-tuned multi-task model performs better than the standard multi-task model, similar to the Finnish experiments. On the transcripts generated by the models, we can observe that the augmented labels approach achieves the highest F1 score. We can also observe that both the augmented labels and the fine-tuned multi-task approaches outperform the pipeline approach.

Table 6. WER on the Swedish test set.

Model	WER
Baseline ASR	33.44
AL	33.82
MT	34.58

Table 7. Precision, recall and F1 score for the Swedish test set.

Transcripts	Model	Prec	Rec	F1
Original	MT	97.76	91.27	94.40
	MT*	98.32	93.48	95.84
Generated	Pipeline	69.35	79.37	74.02
	AL	74.96	78.13	76.51
	MT	70.14	77.94	73.83
	MT*	74.19	76.67	75.41

5.3 English Results

Next, we present the results obtained on the English data sets. In Table 8, we can see how our models perform in terms of WER when evaluated on the LibriSpeech and the English-Gold test sets. From the table, we can see that both approaches perform slightly better than the baseline ASR model trained on the LibriSpeech data. On the English-Gold, on the other hand, the multi-task model performs slightly better than the baseline, whereas the augmented labels yield worse results. Additionally, we can see that on the Libri clean test set, both approaches perform really close, whereas on the Libri other test set, the multi-task approach performs slightly better. Additionally, the multi-task

approach performs better than the augmented labels on the English-Gold test set as well.

On the NER task, when evaluated on the original transcripts, the fine-tuned multi-task approach outperforms the normal multi-task approach on all the English data sets. On the transcripts generated by the models, we can see that the pipeline approach is better than our proposed E2E models on the LibriSpeech test sets. On the manually annotated English Gold test set, on the other hand, the multi-task approach achieves the best F1 score. Additionally, both the multi-task and the augmented labels approaches perform better than the pipeline approach.

Table 8. WER on the LibriSpeech and English-Gold test sets.

Model	Libri clean	Libri other	English-Gold
Baseline ASR	12.74	31.61	23.26
AL	12.34	30.88	23.51
MT	12.35	30.56	23.07

Table 9. Precision, recall and F1 score for the English test sets.

Transcripts	Model	Libri clean			Libri other			English Gold		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Original	MT	87.82	86.01	86.90	86.95	86.23	86.59	64.44	77.09	70.20
	MT*	88.41	86.46	87.43	87.55	86.13	86.83	81.86	68.02	74.30
Generated	Pipeline	76.43	79.09	77.74	64.07	74.40	68.85	79.24	71.28	75.05
	AL	79.77	63.47	70.69	70.21	52.15	59.85	82.60	69.30	75.21
	MT	74.63	76.77	75.68	60.90	73.44	66.59	77.04	84.89	80.78
	MT*	76.33	77.10	76.72	63.33	71.75	67.29	81.86	68.02	74.30

6 Analysis of the Results

To further investigate the NER performance of the models, we plotted confusion matrices. In Figure 1, we can see how the augmented labels and fine-tuned multi-task approaches perform on individual named entity classes on the Finnish data set. We can notice from the confusion matrices that both approaches are doing a pretty good job at detecting the entities, especially the location. On the other hand, they sometimes confuse non-entities with entities. This is especially visible in the person and organization classes, where some non-entities are tagged with either of them.

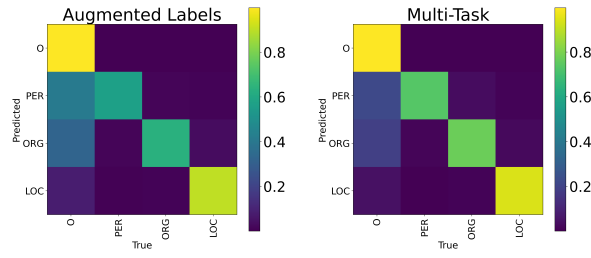


Fig. 1. Confusion matrices for the AL and MT* models, evaluated on the transcripts generated by the models, using the Finnish parliament test set.

Similar to the Finnish results, in Figure 2, we can observe that on the Swedish data set, the models do not have difficulties recognizing the entities. Furthermore, we can see that in a small number of cases, the models confuse the person entity with a location. Additionally, we can see that most of the mistakes that the models make are by confusing non-entities with entities, just like in the Finnish results.

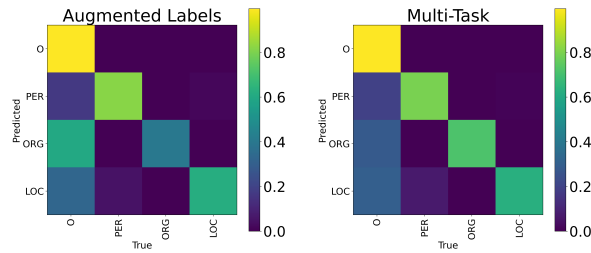


Fig. 2. Confusion matrices for the AL and MT* models, evaluated on the transcripts generated by the models, using the Swedish test set.

On the English-Gold test set, as shown in Figure 3, we can observe that the models make more mistakes than on the other data sets. That is especially the case with the organization entity. The reason for that could be because there are far fewer organization entities in the LibriSpeech and English-Gold data sets, in comparison to the other entities. To ensure that the bad recognition score for the organization entity is expected, we additionally compared the score to the one obtained by the pipeline model. When evaluated on the test data, the pipeline approach also got a low score for the organization entity. Generally, since the English-Gold data set is a combination of many different data sets, it is expected that the domain mismatch negatively impacts the NER.

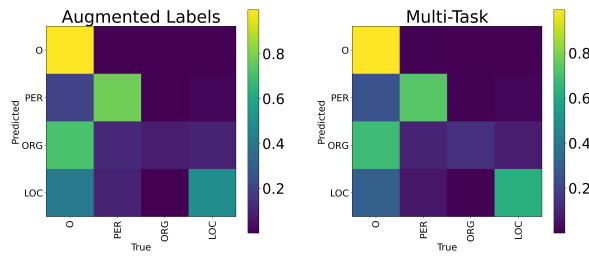


Fig. 3. Confusion matrices for the AL and MT* models, evaluated on the transcripts generated by the models, using the English-Gold test set.

In terms of model complexity, the multi-task approach has more parameters than the augmented labels. For instance, the multi-task model trained on the LibriSpeech data has 22.3 million parameters, in comparison to the augmented labels model, which has 19.6 million parameters. This is expected due to the fact that the multi-task approach has an additional NER branch, which adds to the model complexity.

7 Conclusion

In this paper, we presented two approaches for end-to-end named entity recognition and evaluated them on Finnish, Swedish, and English data sets. We showed that both approaches perform similarly in terms of WER, against the baseline model. Even though the WER results are not in pair with the current state of the art, the goal of this paper is to show that named entities can be learned in an E2E manner, without sacrificing too much of the ASR performance. This allows the ASR part to be optimized for the NER task and vice versa. In terms of the F1 score, both approaches achieve promising results. When comparing both systems, the multi-task approach outperforms the augmented labels approach on the NER task by a significant margin, in all the experiments, except the Swedish, when evaluated on the transcripts generated by the models. When compared against the standard pipeline approach, our proposed models achieve better results on most of the experiments. Generally, we can say that the multi-task approach is more flexible, allowing us to additionally fine-tune the NER branch, which gives an improvement in almost all the experiments. In the future, we plan to replace the models with a Transformer architecture and see how it performs in comparison to the BLSTMs.

References

1. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al.: Deep speech 2: End-to-end speech recognition in english and mandarin. In: International conference on machine learning. pp. 173–182 (2016)
2. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Advances in neural information processing systems. pp. 577–585 (2015)

3. Deoras, A., Sarikaya, R.: Deep belief network based semantic taggers for spoken language understanding. In: Interspeech. pp. 2713–2717 (2013)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A., Morin, E.: End-to-end named entity and semantic concept extraction from speech. In: 2018 IEEE Spoken Language Technology Workshop (SLT). pp. 692–699. IEEE (2018)
6. Gravano, A., Jansche, M., Bacchiani, M.: Restoring punctuation and capitalization in transcribed speech. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 4741–4744. IEEE (2009)
7. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376 (2006)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
9. Jeong, M., Lee, G.G.: Jointly predicting dialog act and named entity for spoken language understanding. In: 2006 IEEE Spoken Language Technology Workshop. pp. 66–69. IEEE (2006)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001). pp. 282–289 (2001)
12. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
13. Malmsten, M., Börjeson, L., Haffenden, C.: Playing with words at the national library of sweden – making a swedish bert (2020)
14. Mansikkaniemi, A., Smit, P., Kurimo, M., et al.: Automatic construction of the finnish parliament speech corpus. In: INTERSPEECH. vol. 8, pp. 3762–3766 (2017)
15. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Interspeech. pp. 3771–3775 (2013)
16. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5206–5210. IEEE (2015)
17. Porjazovski, D., Leinonen, J., Kurimo, M.: Named entity recognition for spoken finnish. In: Proceedings of the 2nd International Workshop on AI for Smart TV Content Production, Access and Delivery. pp. 25–29 (2020)
18. Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
19. Toshniwal, S., Kannan, A., Chiu, C.C., Wu, Y., Sainath, T.N., Livescu, K.: A comparison of techniques for language model integration in encoder-decoder speech recognition. In: 2018 IEEE spoken language technology workshop (SLT). pp. 369–375. IEEE (2018)
20. Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T.: Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing* **11**(8), 1240–1253 (2017)
21. Yadav, H., Ghosh, S., Yu, Y., Shah, R.R.: End-to-end named entity recognition from english speech. arXiv preprint arXiv:2005.11184 (2020)
22. Zhai, L., Fung, P., Schwartz, R., Carpuat, M., Wu, D.: Using n-best lists for named entity recognition from chinese speech. In: Proceedings of HLT-NAACL 2004: Short Papers. pp. 37–40 (2004)