MeMAD

Methods for Managing Audiovisual Data

memad.eu
info@memad.eu

Twitter – @memadproject
LinkedIn – MeMAD Project

MeMAD Deliverable

D3.2 TV moments detection and linking (initial 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	08.05.2019
Start date of the project	1.1.2018
Due date of the deliverable	31.12.2019
Actual date of submission	31.12.2019, (*updated on*) 10.12.2020
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 theMeMAD 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
Mokhles Bouzaien	EURECOM	mokhles.bouzaien@eurecom.fr
Ismail Harrando	EURECOM	ismail.harrando@eurecom.fr
Benoit Huet	EURECOM	benoit.huet@eurecom.fr
Mikko Kurimo	AALTO	mikko.kurimo@aalto.fi
Tiina Lindh-Knuutila	LLS	tiina.lindh-knuutila@lingsoft.fi
Dejan Porjazovski	AALTO	dejan.porjazovski@aalto.fi
Alison Reboud	EURECOM	alison.reboud@eurecom.fr
Michael Stormbom	Lingsoft	michael.stormbom@lingsoft.fi
Raphaël Troncy	EURECOM	raphael.troncy@eurecom.fr

Internal reviewers in alphabetical order		
Name	Beneficiary	e-mail
David Doukhan	INA	ddoukhan@ina.fr
Dieter Van Rijsselbergen	Limecraft	dieter.vanrijsselbergen@limecraft.com

Abstract

This deliverable further describes the MeMAD Knowledge Graph that has been initiated in the Deliverable D3.1. In particular, the MeMAD Knowledge Graph has been greatly enhanced by integrating the first results of the multimodal media analysis performed in WP2, such as the ASR and face recognition results obtained on medias from the MeMAD corpora. The MeMAD knowledge graph mostly re-uses the EBU Core ontology but it also makes use of several other ontologies such as the W3C Media Annotations, W3C Media Fragments, W3C Web Annotations vocabularies. We have participated in extending the EBU Core ontology and we are now listed as contributors since the latest version includes all MeMAD proposals for extension, in particular for modeling results of automatic analysis. We also proposed a new approach that amounts to merge SPARQL bindings on the base of identifiers and the integration in the grlc API framework to create new bridges between the Web of Data and the Web of applications. This has lead to a significant research result materialized by the development of the SPARQL Transformer library available in Javascript and in Python. This library is used to automatically generate the MeMAD knowledge graph API which is then used by the Limecraft Flow platform. As another client using the API built on top of the knowledge graph, we unveiled a new web application named MeMAD Explorer that provides an exploratory search engine for the MeMAD corpora.

This deliverable mostly describes 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 and interestingness. We relied on different standard datasets that provide such annotations for training and testing our methods, namely the MediaEval Predicting Media Memorability dataset, the MediaEval Predicting Media Interestingness dataset, and the Cognimuse Media Interestingness dataset. We have conducted numerous experiments including using very recent visio-linguistic models such as VilBERT. We discuss the current results as well the still open research problem for being able to generalize the current methods across video genres and themes.

Once important moments are detected from videos, we aim to enrich those with additional in-

Contents

1 Introduction **5**

2 MeMAD Knowledge Graph **6**
2.1 Modeling the Knowledge Graph . . . 6
2.2 Semantic annotations for media content . . . 6
2.3 A Program Description in the Knowledge Graph . . . 9
2.4 Knowledge Graph Statistics . . . 11
2.5 Providing access to the MeMAD Knowledge Graph via a REST API . . . 11
2.6 Browsing the MeMAD programs in an Exploratory Search Engine . . . 14

3 Predicting Media Memorability and Interestingness **17**
3.1 Related Work on Visio-Linguistic Models . . . 17
3.1.1 Transformers . . . 17
3.1.2 BERT . . . 18
3.1.3 ViLBERT . . . 18
3.1.4 VisDial-BERT . . . 19
3.1.5 VisualBERT . . . 19
3.1.6 VideoBERT . . . 20
3.1.7 VL-BERT . . . 20
3.2 Predicting Media Memorability . . . 20
3.2.1 Related Work on Important Moment Detection . . . 21
3.2.2 Approach 1: Combining Visual and Textual Features to Predict Media Memorability . . . 22
3.2.3 Approach 2: Using Visio-Linguistic Models to Predict Media Memorability 24
3.2.4 Results and Analysis . . . 28
3.3 Predicting Media Interestingness . . . 29
3.3.1 Motivation . . . 29
3.3.2 Model Architecture . . . 30
3.3.3 Experiments on the Cognimuse database . . . 31
3.3.4 Experiments on the MediaEval 2017 dataset . . . 31

4 Moments Enrichment **33**
4.1 NER methodologies . . . 33
4.1.1 Aalto NER . . . 33
4.1.2 Methods . . . 34
4.1.3 Lingsoft Analyzer with semantic linking . . . 34
4.2 Finnish NER benchmarking results . . . 37
4.2.1 Finnish FiNER test set . . . 37
4.2.2 Extending Yle MeMAD broadcast media evaluation set for NER in Finnish and Swedish . . . 39
4.2.3 Benchmarking results with the Yle MeMAD broadcast evaluation set for NER . . . 39
4.3 NER on ASR . . . 41
4.3.1 Data . . . 41
4.3.2 Results . . . 42
4.3.3 Conclusion . . . 43
4.4 Fine-grained Named Entity Recognition . . . 43
4.4.1 TAC-KBP Entity Discovery and Linking challenge . . . 43

4.4.2 Ontonotes dataset . . . 44
4.4.3 Approach . . . 44
4.4.4 The model . . . 45
4.4.5 Results . . . 46
4.5 EURECOM NER on French ASR . . . 47
4.5.1 Breakdown by Publication Channel and Entity Type . . . 48
4.5.2 Breakdown by Genre . . . 48
4.5.3 A closer look . . . 49
4.5.4 Wikifier output . . . 50
4.6 Aligning ASR with manual subtitles . . . 50
4.6.1 The alignement process . . . 50
4.6.2 NER on the aligned corpora . . . 51

5 Conclusion **52**

6 References **53**

A Dissemination activities **58**

B Appendices **59**
B.1 EURECOM MDN 2019 Talk . . . 59
B.2 EURECOM and AALTO's MediaEval 2019 workshop paper [1] . . . 59
B.3 EURECOM ISWC 2019 conference paper [2] . . . 63
B.4 Aalto's ACM AI4TV 2020 paper . . . 81

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 interoperation 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 broadcast archives, web media, news and photo stock agencies or social networks.

The general methodology that we follow consists in: 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 true 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 consolidated the MeMAD Knowledge Graph that was initiated during the first year of the project. In particular, we have further integrated the results of the multimodal media analysis performed in WP2, namely all textual resources associated to the medias (subtitles, automatic speech recognition, teleprompters) and identification of celebrities visually recognized in the video. From the knowledge engineering side, we are pleased to have contributed to the latest release of the EBU Core ontology, which now includes suggestions made by MeMAD for annotating broadcasts with results from multimedia automatic analysis. We also research and develop a novel method for automatically generating a REST-based API to any knowledge graph. The resulting MeMAD API is being used to integrate the content provider legacy metadata in the Flow platform as well as a rich client named MeMAD Explorer (Section 2).

We developed novel 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 and interestingness. We relied on different standard datasets that provide such annotations for training and testing our methods, namely the MediaEval Predicting Media Memorability dataset, the MediaEval Predicting Media Interestingness dataset, and the Cognimuse Media Interestingness dataset. We have conducted numerous experiments including using very recent visio-linguistic models such as VilBERT. We discuss the current results as well the still open research problem for being able to generalize the current methods across video genres and themes (Section 3).

Once important moments are detected from videos, we aim to enrich those with additional information. These enrichments are typically information coming from encyclopedia that would further describe a named entity or a technical concept mentioned in the program, or it can be another video moments coming from the same broadcast or from another broadcast that is related. In this deliverable, we report on three different methods for extracting and disambiguating named entities from textual resources associated to videos, for some common types (person, organization, location, etc.) and some languages (English, French, Finnish, Swedish) and using Wikidata as background knowledge. These methods have been successfully applied to the MeMAD datasets generating competitive results. These annotations are enrichment candidates that can be proposed to the end users (Section 4).

2 MeMAD Knowledge Graph

In this section, we describe the evolution of the MeMAD Knowledge Graph following the initial version released in the Deliverable D3.1 [3]. This includes the conversion of the legacy metadata (Section 2.1) and the modeling of annotations coming from automatic analysis (Section 2.2). We provide a full program description example in Section 2.3 and more stats of the Knowledge Graph in Section 2.4. One of the key research result for this second year of the project is the development of a generic solution for providing a generic but configurable RESTful API on top of any RDF-based Knowledge Graph, and in particular for the MeMAD one. We describe this method in the Section 2.5. This API aims to facilitate the development of rich web applications that can access the knowledge graph. We illustrate such a rich application with the preliminary version of the MeMAD Explorer, an exploratory search engine for browsing the MeMAD corpora based on the legacy metadata (Section 2.6).

2.1 Modeling the Knowledge Graph

Following the work done during the first year of the project, we further integrate all metadata supplied by the content providers into the MeMAD Knowledge Graph (Figure 1). This includes the conversion of new datasets from Yle, unification of date and duration formats across datasets, and the integration of all timed textual data into the knowledge graph, i.e. subtitles from Yle, automatically-generated transcripts and teleprompter feed for news programs from INA. There were also some significant work done to correct some of the problems emerging from the conversion process and the unification of metadata representation across the datasets, i.e. using relative times to express the starting time of a segment inside of a program (while this information is generally provided in the data in absolute time), explicitly expressing the start time, end time and duration of each segment (usually either one of the last two is omitted), annotating the first broadcast information of every program, etc.

The updated conversion scripts for all datasets are available on the project Github repository at `https://github.com/MeMAD-project/rdf-converter`.

2.2 Semantic annotations for media content

As explored in more details in the Deliverable D6.4 [4], we designed interchange formats to represent the results of automatic analysis processes corresponding to the different use cases. They are all expressed in RDF, and mostly leverage elements from existing ontologies such as `ebucore:TextLine`[1] and `nif:Annotation`[2].

Since the primary results on Named Entity Identification and Linking as well as Facial Recognition are already available, we worked on fleshing out these annotations and adding them to the knowledge graph. To make the process of retrieving these annotations from the Knowledge Graph, we endow them with new classes added to the MeMAD ontology e.g. `memad:VisualPersonIdentification`.

```
<http://data.memad.eu/media/UUID1#t=npt:2898.000000&xywh=213,75,54,76> a
   ebucore:MediaFragment ;
    ebucore:isMediaFragmentOf <http://data.memad.eu/media/UUID1> .

<http://data.memad.eu/annotation/UUID2> a oa:Annotation ;
```

[1] `https://www.ebu.ch/metadata/ontologies/ebucore/`
[2] `https://github.com/NLP2RDF/ontologies/blob/master/nif-core/nif-core.ttl`

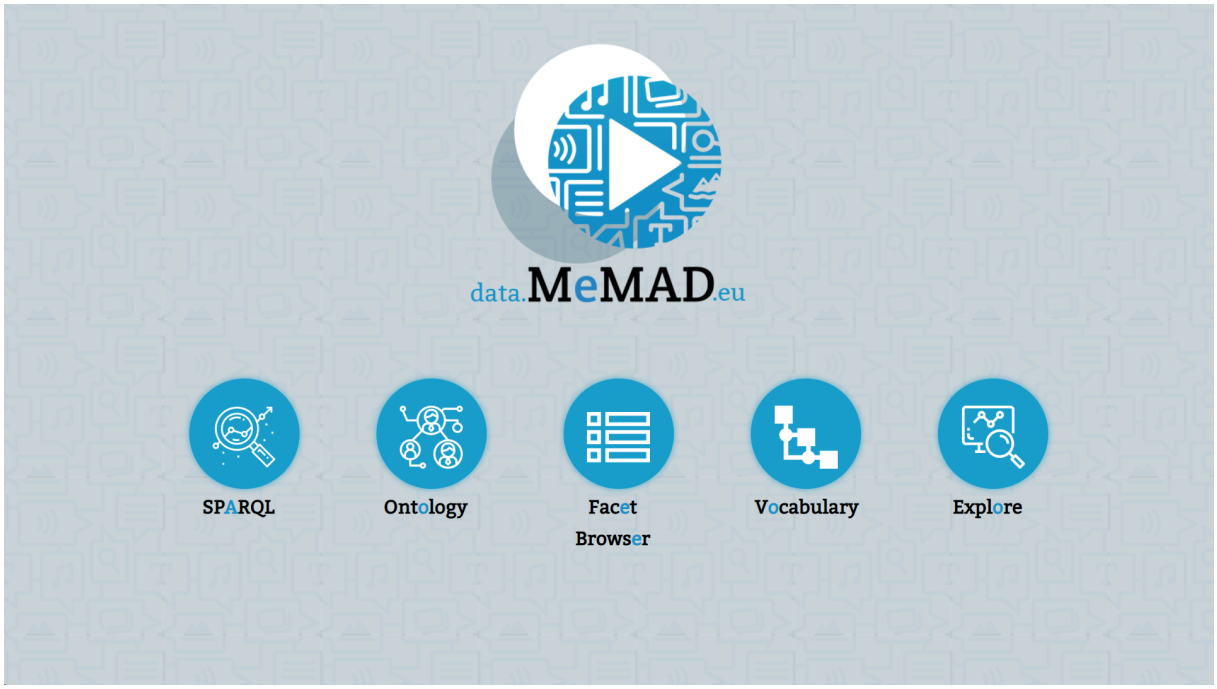

Figure 1: The landing page of the MeMAD data platform at `http://data.memad.eu/` offering access to the MeMAD ontology, a protected SPARQL endpoint and facetted browser over the knowledge graph, and the MeMAD API

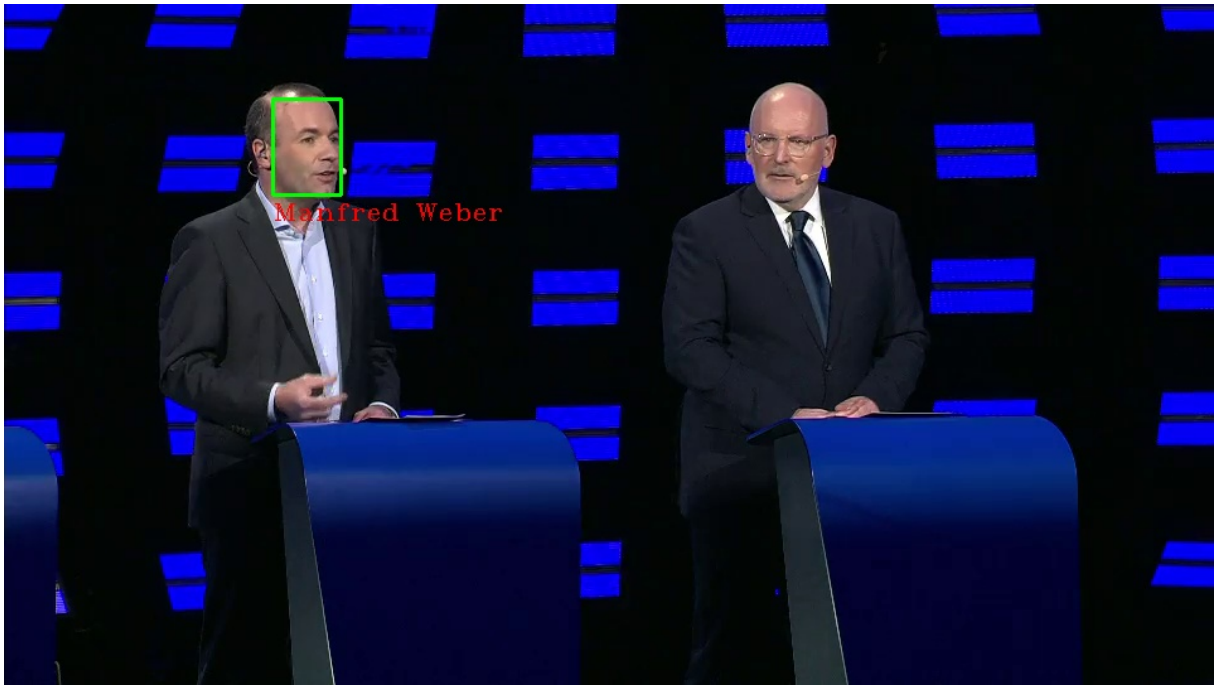

Figure 2: An example of a celebrity face recognition result

```
    dcterms:created "2019-12-13T12:09:39.137201"^^xsd:datetime ;
    dcterms:creator <http://data.memad.eu/organization/EURECOM> ;
    dcterms:motivatedBy oa:identifying ;
    oa:hasTarget
        <http://data.memad.eu/media/UUID1#t=npt:2898.00&xywh=213,75,54,76> ;
    oa:hasBody <http://data.memad.eu/yle/eurovaalit2019/UUID1/annotation/1> .

<http://data.memad.eu/yle/eurovaalit2019/UUID1/annotation/1> a nif:Annotation,
    memad:VisualPersonIdentification ;
    nif:taIdentConf "0.75499" ;
    nif:taIdentProv "EURECOM Celebrity Face Recognition" ;
    rdf:value "Manfred Weber" ;
    itsrdf:taIdentRef <https://www.wikidata.org/wiki/Q88866> .
```

Listing 1: Celebrity Face Recognition results, expressed in RDF

We provide a complete example of video annotation (at the frame level) with results from celebrity face recognition (Figure 2 and Listing 1) and a complete example of named entity recognition and disambiguation on subtitles (Listing 2). In both cases, we identify the object to annotate (`ebucore:MediaFragment`, `nif:String`) as well as how it's attached to the original media resource / editorial object (`ebucore:isMediaFragmentOf`, `ebucore:hasRelatedTextLine` / `ebucore:textLineContent`). For readability, we omit the actual program identifier (40 characters-long hashes of internal identifiers) and use `UUID`n instead.

```
<http://data.memad.eu/media/UUID3#t=2679.49,2745.09> a ebucore:MediaFragment ;
    ebucore:isMediaFragmentOf <http://data.memad.eu/media/UUID3> .

<http://data.memad.eu/fcr/les-matins-de-france-culture/UUID3/textline/9> a
    ebucore:TextLine ;
    ebucore:textLineContent
        <http://data.memad.eu/fcr/les-matins-de-france-culture/UUID3/
     textline/9/content> ;
    ebucore:textLineStartTime "00:44:39.49"^^xsd:time ;
    ebucore:textLineEndTime "00:45:45.09"^^xsd:time ;
    ebucore:textLineLanguage "fr"^^xsd:language .

<http://data.memad.eu/fr2/8h00-le-journal/UUID3/textline/9/content>
    a nif:String, nif:Context ;
    nif:isString "Marine Le Pen se laisse offrir le luxe d'une couv du Times
        de portraits et d'interviews démultiplié àl'infini , chez nos voisins
        européens , je suis partout , semble-t-elle nous dire et surtout
        ailleurs , ce qui n'est pas le moindre des paradoxes de cette échappée
        ." .

<http://data.memad.eu/fcr/les-matins-de-france-culture/UUID3/textline/9/
 annotation/1>
    a nif:OffsetBasedString, nif:Annotation, nif:EntityOccurrence ;
    nif:referenceContext
        <http://data.memad.eu/fr2/8h00-le-journal/UUID3/textline/9/content> ;
```

```
    nif:beginIndex "0"^^xsd:integer ;
    nif:endIndex "13"^^xsd:integer ;
    nif:anchorOf "Marine Le Pen"^^xsd:string ;
    # NER results
    itsrdf:taClassRef nerd:Person ;
    nif:taClassConf "0.95"^^xsd:decimal ;
    nif:taIdentProv "DeepNER" ;
    # NED results
    itsrdf:taIdentRef <https://www.wikidata.org/wiki/Q12927> ;
    nif:taIdentConf "0.9"^^xsd:decimal ;
    nif:taIdentProv "ADEL" ;
    itsrdf:taSource "Wikidata" .

<http://data.memad.eu/annotation/UUID4> a oa:Annotation ;
    dcterms:creator <http://data.memad.eu/organization/EURECOM> ;
    dcterms:created "2019-01-20"^^xsd:date ;
    dcterms:motivatedBy oa:classifying, oa:identifying ;
    oa:hasTarget <http://data.memad.eu/media/UUID3#t=2679.49,2745.09>
    oa:hasBody <http://data.memad.eu/fcr/les-matins-de-france-culture/UUID3/
     textline/9/annotation/1> .
```

Listing 2: NER Results, expressed in RDF

2.3 A Program Description in the Knowledge Graph

The Figure 3 provides an example for the main properties of a program once converted into RDF, as well as all the main classes being used. It describes a `ebucore:RadioProgram` identified in the knowledge graph by `http://data.memad.eu/fit/inter-soir-18h00/76c9e5b10d14c900787178f64a67a0591d848671`, broadcasted on *19/05/2014*. The title of the program is *Inter soir 18h00 : émission du 19 mai 2014* of genre `Journal parlé`. This program belongs to a collection identified by `http://data.memad.eu/fit/inter-soir-18h00`. Ina's archivists have further described two particular segments of this program. Those are typed as `ebucore:Part` in the knowledge graph. One of them, identified by `http://data.memad.eu/fit/inter-soir-18h00/65695aea6f585fefe9dc244a023d63e1928eb8c1`, has for title *Festival de Cannes : projection en compétition de "L'institutrice" de l'Israélien Nadav Lapid* and for summary *Commentaires d'Eva Bettan. Itw de Nadav LAPID, réalisateur : enfant, il écrivait des poèmes dès l'âge de 4 ans, qui apparaissent dans le film. Il est l'enfant, mais il est aussi, plutôt l'institutrice. On veut sauver cet enfant de la vulgarité du monde, en sachant qu'il est impossible de lutter contre l'esprit du temps.*. The documentalists have identified three remarkable persons as contributors. In the knowledge graph, we mint new URIs for those people that are further disambiguated. Hence, the person `Nadav Lapid` is identified by `http://data.memad.eu/agent/lapid-nadav` and could be stated as being `owl:sameAs` than `https://www.wikidata.org/wiki/Q7060835`. Clicking on this URI immediately provides other programs in which this person appeared, e.g. on other radio programs from other channels such as France Culture at the time of the Cannes Film Festival where he was promoting his last movie. It should be noted that for most entities in the Knowledge Graph, not all properties are necessarily valued.

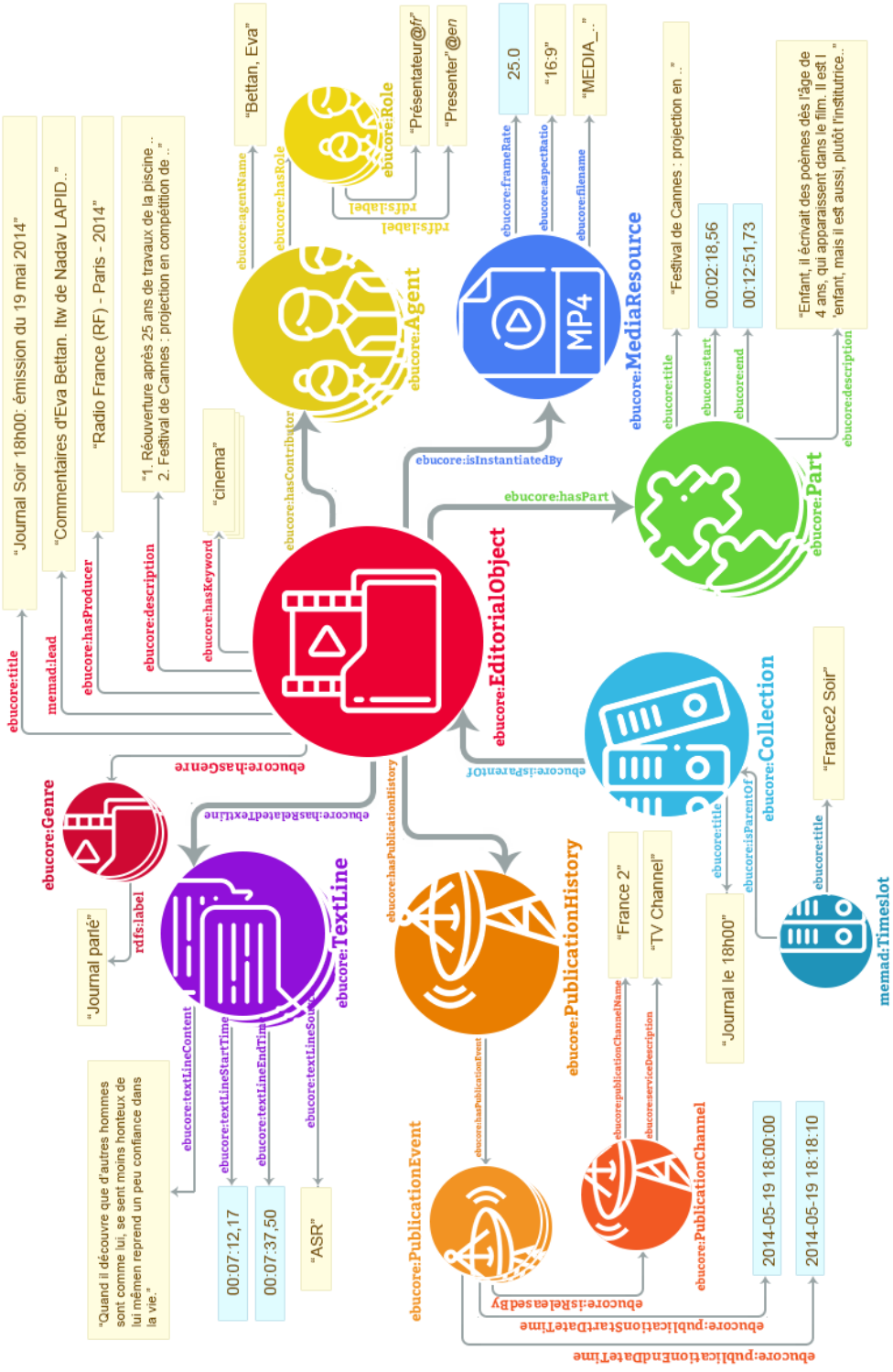

Figure 3: An example of the output of the RDF conversion. All URIs are dereferencable but behind an access control layer since some metadata cannot be exposed publicly.

2.4 Knowledge Graph Statistics

At the end of the conversion process, we study the content of the knowledge graph for each dataset that has been ingested. Tables 1, 2, and 3 present the breakdown of the main properties' coverage for INA's Legal Deposit, INA's Professional Archive and Yle's datasets, respectively. While each program always come with a title, a genre and some publication events, the remaining of the legacy metadata vary. We observe a relatively high coverage for properties such as contributing agents and summaries. However, the coverage for the rest of the properties is relatively sparse. This triggers opportunities for automatic multimedia analysis that can complement this manual documentation in predicting values for some properties when being absent (e.g. the genre of a programme).

Property	**TV Instances**	**TV Coverage**	**Radio Instances**	**Radio Coverage**
Programmes	236		852	
Title	236	100%	852	100%
Genre	236	100%	852	100%
Publication Channel	236	100%	852	100%
Summary	207	87.71%	157	18.43%
Producer Summary	0	0%	515	60.45%
Themes	135	57.2%	414	48.59%
Keywords	100	42.37%	305	35.8%
Producers	146	61.86%	852	100%
Contributors	208	88.14%	640	75.12%
Collection	209	88.56%	311	36.5%

Table 1: Number of instances and property coverage from INA's Professional Archive

Property	**TV Instances**	**TV Coverage**	**Radio Instances**	**Radio Coverage**
Programmes	86789		21440	
Title	86789	100%	21440	100%
Genre	86789	100%	21403	99.83%
Publication Channel	86789	100%	21403	99.83%
Summary	506	0.58%	420	1.96%
Lead	1972	2.27%	2049	9.56%
Producer Summary	13326	15.35%	6645	30.99%
Themes	70310	81.01%	4334	20.21%
Keywords	4021	4.63%	2685	12.52%
Producers	75480	86.97%	16555	77.22%
Contributors	12503	14.41%	6645	30.99%
Collection	75480	86.97%	19682	91.8%

Table 2: Number of instances and property coverage from INA's Legal Deposit

2.5 Providing access to the MeMAD Knowledge Graph via a REST API

The MeMAD knowledge graph can be accessed via a protected SPARQL endpoint at `http://data.memad.eu/sparql`. It requires an authentication since the content providers do not wish that all metadata are publicly exposed. In addition, all URIs are also linked data URIs, i.e. dereferencable and providing an RDF description when being HTTP GET.

In a document-based world as the one of Web APIs, the triple-based output of SPARQL endpoints can, however, be a barrier for developers who want to integrate Linked Data in their applications. A different JSON output can be obtained with SPARQL Transformer [5], which relies on a single JSON object for defining which data should be extracted from the

Property	Instances	Coverage
TV Programmes	4335	
Title	4335	100.0%
Genre	4335	100.0%
Publication Channel	4335	100.0%
Description	3779	87.17%
Themes	4025	92.85%
Subject	1574	36.31%
Subtitles	481	11.1%
Contributors	2840	65.51%
Collection	4013	92.57%

Table 3: Number of instances and property coverage from Yle

endpoint and which shape should they assume. During this review period, we research and propose a new approach that amounts to merge SPARQL bindings on the base of identifiers and the integration in the `grlc` API framework to create new bridges between the Web of Data and the Web of applications [2].

We use this framework to develop the MeMAD API. The API calls being defined are then used by the Flow platform to integrate the MeMAD knowledge graph in the production environment (Figure 4). It works under a simple principle: all one needs to do is to define a *prototype* to the kind of objects the API call should return. This prototype is either a plain JSON object or a JSON-LD one that lists the properties to probe as well as the anchor entity to which all these properties are linked. Once these prototypes defined appropriately, they can be pushed into a public repository and used as a scaffold for the API.

Figure 4: A special 3 hours long TV program dedicated to the 2014 European Elections results broadcasted on the French France 2 channel on prime time, on 25/05/2014 at 19:40. The media is provided by Ina and ingested in the Flow platform. The metadata being displayed comes from the MeMAD knowledge graph.

To illustrate this idea, we provide the API definition for *program_metadata*, an API that, given the internal filename corresponding to a program's media resource, returns its metadata in an aggregated, ready-to-use JSON object. Thus, the Flow platform, after ingesting media files

from the content providers, can use the filenames to query the Knowledge Graph for all the metadata it needs to describe it, with minimal effort (Figure 4).

```
{
  "proto":
    {
      "uri": "?uri$anchor",
      "filename" : "$ebucore:isInstantiatedBy / ebucore:filename$required$var:_filename",
      "identifier" : "$ebucore:hasIdentifier",
      "title" : "$ebucore:title$required",
      "description": "$ebucore:description",
      "summary": "$ebucore:summary",
      "episodeNumber" : "$ebucore:episodeNumber",
      "tags": "$ebucore:hasKeyword",
      "genre": "$ebucore:hasGenre",
      "languages" : "$ebucore:hasLanguage",
      "mainTitle" : "$ebucore:mainTitle",
      "workingTitle" : "$ebucore:workingTitle",
      "theme" : "$ebucore:hasTheme",
      "producer":"$ebucore:hasProducer",
      "publicationChannel" : "$ebucore:hasPublicationHistory / ebucore:hasPublicationEvent /
          ↪ ebucore:isReleasedBy / ebucore:publicationChannelName",
      "firstPublicationTime" : "?startDateTime"
    },
  "$where": [
    "?uri ebucore:hasPublicationHistory / ebucore:hasPublicationEvent ?firstrun.
    ?firstrun a memad:FirstRun; ebucore:publicationStartDateTime ?startDateTime."
  ],
  "$prefixes": {
      "ebucore": "http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#",
      "memad": "http://data.memad.eu/ontology#"
  }
}
```

Listing 3: API definition for *program_metadata*

The response from the API, given a filename (e.g. `MEDIA_2019_01603462.mp4`), will look like this:

```
[
  {
    "description": "Kenelle annetaan, kenelt\"a otetaan? Vaalilupausten hintalappuja setvim\"ass\"a
        ↪ entinen kansliap\"a\"allikk\"o Erkki Virtanen, Kreab Helsingin toimitusjohtaja Mikael
        ↪ Jungner, e2 Tutkimuksen johtaja Karina Jutila ja Nordean ekonomisti Olli K\"arkk\"ainen.
        ↪ Juontajana Sakari Sirkkanen. #yleastudio",
    "episodeNumber": "13",
    "filename": "MEDIA_2019_01603462.mp4",
    "firstPublicationTime": "2019-05-16T21:05:14",
    "genre": [
      "Ajankohtainen",
      "Ajankohtaisohjelma",
      "Keskustelu, haastattelu"
    ],
    "identifier": [
      "MEDIA_2019_01603462"
    ],
    "languages": "Finnish",
    "mainTitle": "A-Talk 88800132483",
    "publicationChannel": [
      "Yle TV1",
```

```
      "Yle Areena",
      "TV Finland"
    ],
    "theme": "Yhteiskunnalliset teemat",
    "title": [
      {
        "language": "se",
        "value": "A-Talk"
      },
      "A-Talk"
    ],
    "uri": "http://data.memad.eu/yle/a-talk/104a554d1efb96bdf9bf0922fff737480f611b75"
  }
]
```


Listing 4: JSON-formatted response for the *program_metadata* API

Using the SPARQL-Transformer allows also the definition of a preferred language (in case of queries which can returns multiple values for the same properties), pagination, ordering and grouping of results etc.

The full documentation is available at `https://github.com/D2KLab/sparql-transformer`. The MeMAD API calls are developed in the github repository at `https://github.com/memad-project/api` and automatically deployed at `http://grlc.eurecom.fr/api/memad-project/api#/`.

2.6 Browsing the MeMAD programs in an Exploratory Search Engine

The MeMAD Knowledge Graph integrates all content shared within the project. In order to facilitates access to the program metadata, we built the MeMAD Explorer, an exploratory search engine which gives end-users a visual interface to search through and to interact with the content of the graph.

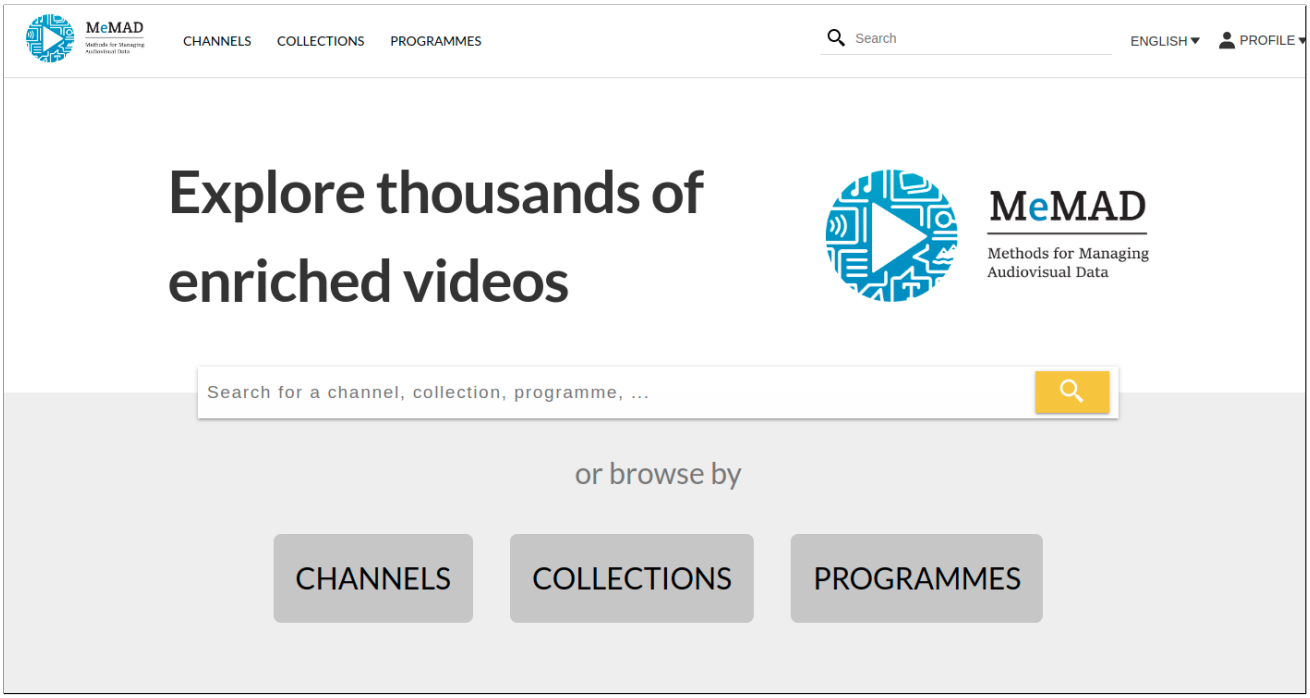

Figure 5: MeMAD Explorer Home Page

The Explorer provides two ways of interacting with the content:

- **The search box**: from the home page (Figure 5), a user can type a query that would be matched with the labels/titles of several objects in the knowledge graph, e.g. programs, collections, channels, etc.
- **The catalog**: the user can browse the catalog of content on the knowledge graph. Through this interface shown in Figure 6, a user can choose through multiple filters to explorer the available content, such as genres, themes, languages and keywords. When logged in (through their Gmail, Facebook or Twitter account), a user can save items from the catalog into a list of favorites to view later.

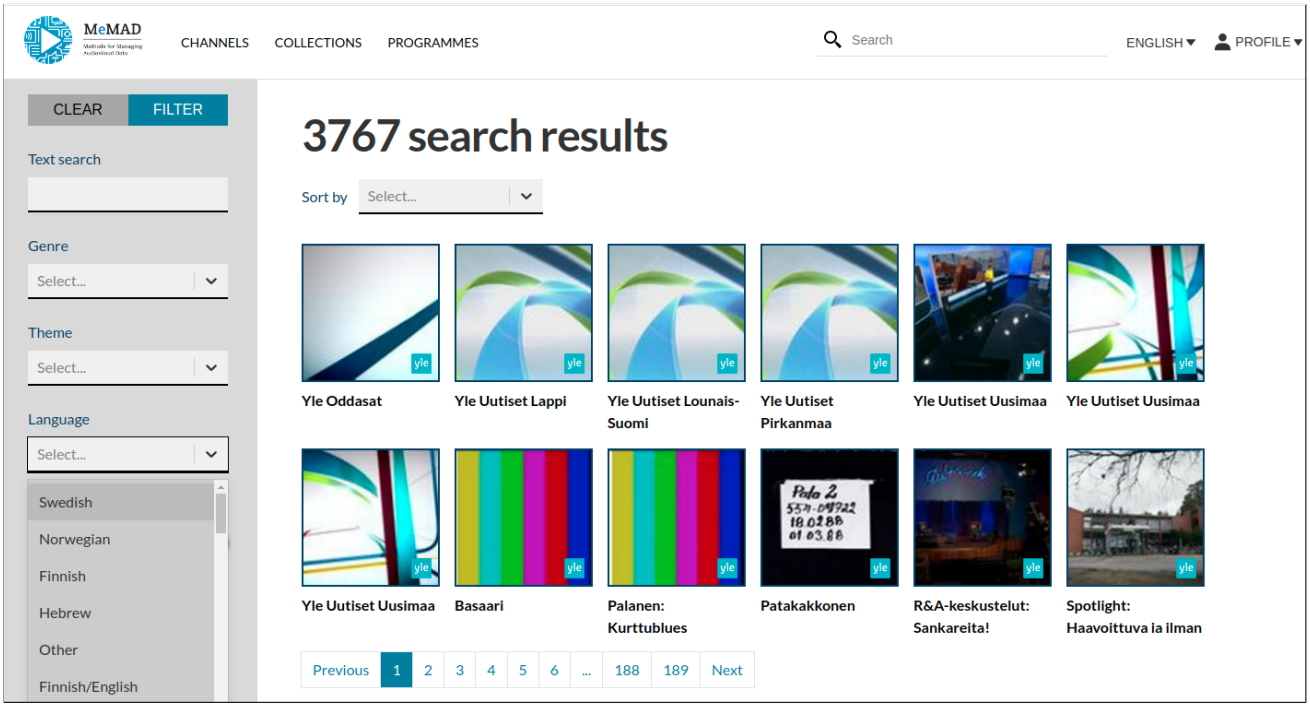

Figure 6: MeMAD Explorer's catalog

When users click on an item, they are directed to the Media Viewer interface (Figure 7) where they can visualize the media content (which is streamed from Limecraft Flow[3], the media hosting and management platform created by Limecraft, a partner in the MeMAD project). On top of that, they can see all the metadata associated with the item, as well as the temporal content segmentation when available, so that they can skip right to the part of the program which is of interest to them.

For the future of the platform, an implementation of the content-based recommendations functionality and the visualization of content enrichment (mentioned entities, face recognition tags..) is planned. The exploratory search engine is available at `http://explorer.memad.eu/` using the credentials `memad` / `memad-pw`.

[3] `https://www.limecraft.com/workflows/media-management/`

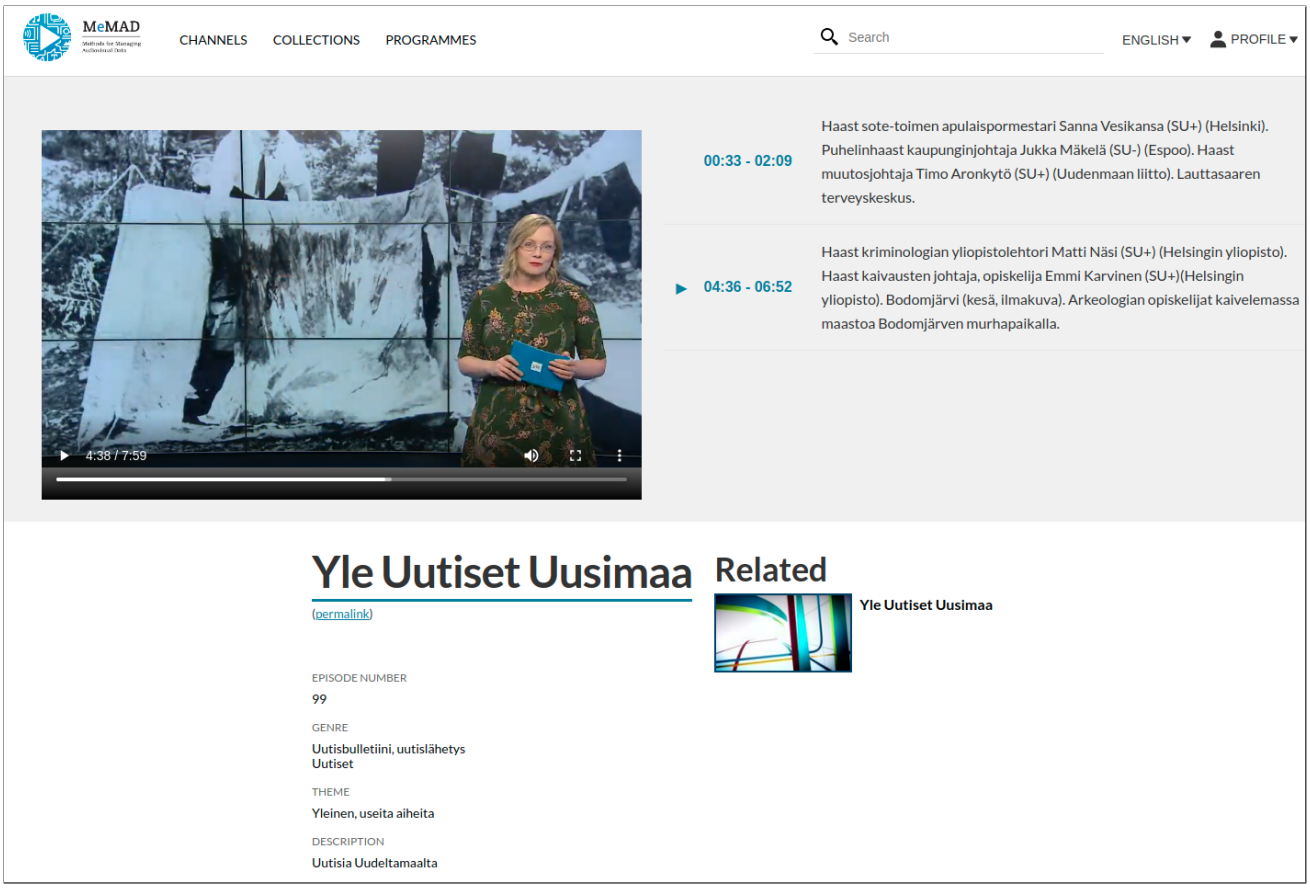

Figure 7: MeMAD Explorer's media viewer

3 Predicting Media Memorability and Interestingness

Radio and TV programs do contain highlights but what makes a highlight is highly subjective. In this deliverable, we refer to this loose concept as an *important moment* which is a sequence of a broadcasted program defined by a start point and an end point that is judged to be important. Since the importance of a video sequence creating such a moment is highly subjective, we consider proxies such as memorability and interestingness. Next, we relied on different standard datasets that provide such annotations for training and testing our methods, namely the MediaEval Predicting Media Memorability dataset, the MediaEval Predicting Media Interestingness dataset, and the Cognimuse Media Interestingness dataset.

In this section, we first provide a state of the art on visio-linguistic models that could help in developing such methods (Section 3.1).

Section 3.2

Section 3.3

We discuss the current results as well the still open research problem for being able to generalize the current methods across video genres and themes.

3.1 Related Work on Visio-Linguistic Models

In 2019, Lu *et al.* have proposed the ViLBERT architecture and model which consists in pretraining task-agnostic visiolinguistic representations for vision and language tasks [6]. This architecture is an extension of the BERT model (which is based on Transformers) that aims to process both textual and visual modalities. In this section, we first review these architectures as well as other related ones such as VisDial-BERT, VisualBERT, VideoBERT and VL-BERT.

3.1.1 Transformers

Transformers were proposed by Vaswani *et al.* in [7]. It is based entirely on an attention mechanism instead of recurrence and convolutions, which reduces the sequential calculations, and makes it parallelizable and faster to train. Before feeding it to the Transformer, the sequence is tokenized. Then, a positional encoding vector is added as detailed in Figure 8.

Encoder. A representation of the sequence is calculated using the self-attention mechanism. It consists of relating different parts of a single sequence by calculating query, key and value vectors ($\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$) for each token using respectively three trainable matrices $\mathbf{W^Q}$, $\mathbf{W^K}$ and $\mathbf{W^V}$. The self-attention mechanism is followed by a normalization layer. Finally, an attention vector is calculated:

$$\text{Attention}\left(\mathbf{q}, \mathbf{k}, \mathbf{v}\right) = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d_k}}\right)\mathbf{v} \tag{1}$$

where d_k is the dimensionality of $\mathbf{k}$. This attention vector represents how much focus to place on other parts of the sequence.

Decoder. The decoder architecture is similar to the one used by the encoder but an extra multi-head attention layer is used over the output of the latter. The last layer maps a float vector to a word.

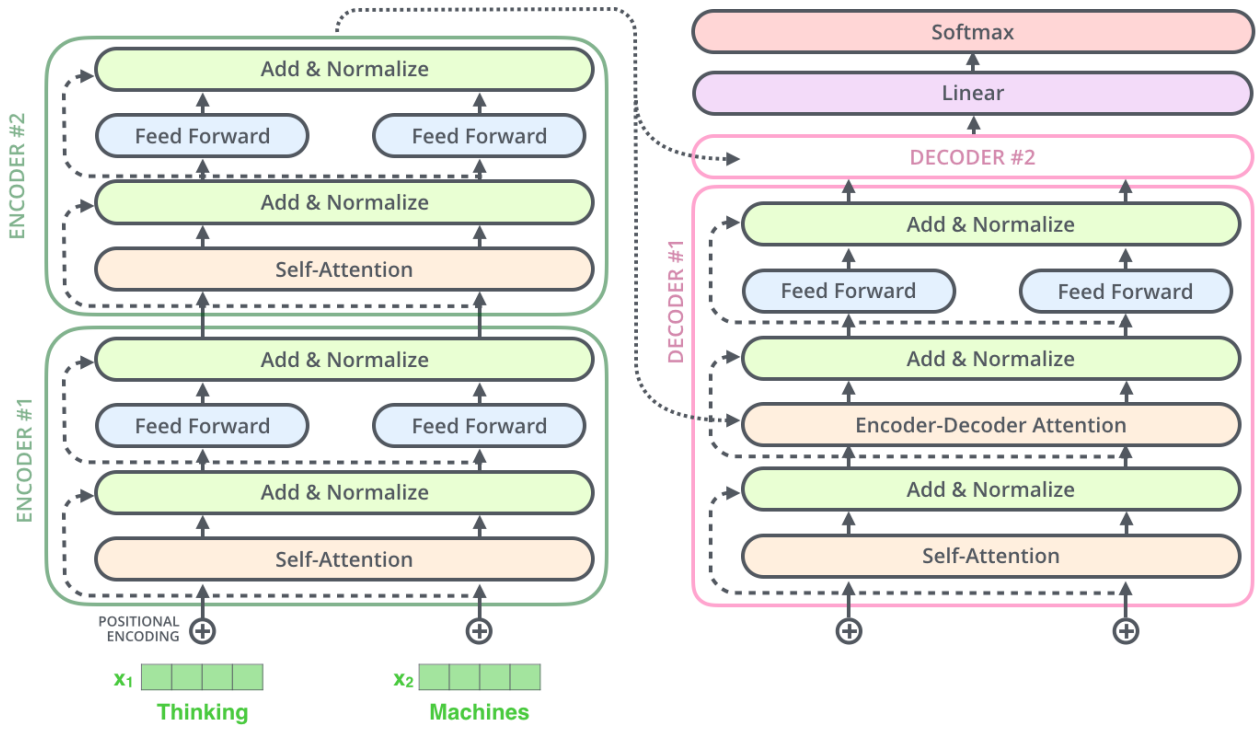

Figure 8: The Transformer - model architecture from [7]

3.1.2 BERT

BERT stands for Bidirectional Encoder Representations from Transformers and was introduced by Devlin et al. in [8]. It is a language representation composed of the concatenation of L Transformer blocks ($L = 12$ or $L = 24$). In each block, a bidirectional self-attention is used, i.e., attention does not only attend to context to the left.

Depending on the task, the model's input can be a single or a pair of tokenized sentences separated with the special token $[SEP]$. Segment and Position Embedding are added to the Token Embedding to constitute the final input. BERT is pre-trained on unlabeled data for two main tasks: Masked Language Modeling and Next Sentence Prediction using BookCorpus (800M words) [9] and English Wikipedia (2.5B words). BERT can be fine-tuned using labeled data to learn specific tasks such as Question Answering [10].

3.1.3 ViLBERT

Architecture. ViLBERT [6] is an extension of BERT which is about learning the associations and links between visual and linguistic properties of a concept that could be a helpful feature for vision-and-language tasks. As shown in Figure 9, ViLBERT has a two-stream architecture modelling each modality (i.e., visual and textual) separately, and then fusing them through a set of attention-based interactions (co-attention). The keys and values of each modality are passed as input to the other modality's multi-headed attention block.

Pre-training and Fine-tuning. ViLBERT is pre-trained using the Conceptual Captions data set (3.3M image-caption pairs) [11] on two main tasks:

- Masked multi-modal learning: the model must reconstruct image region categories or words for masked inputs given the observed inputs.

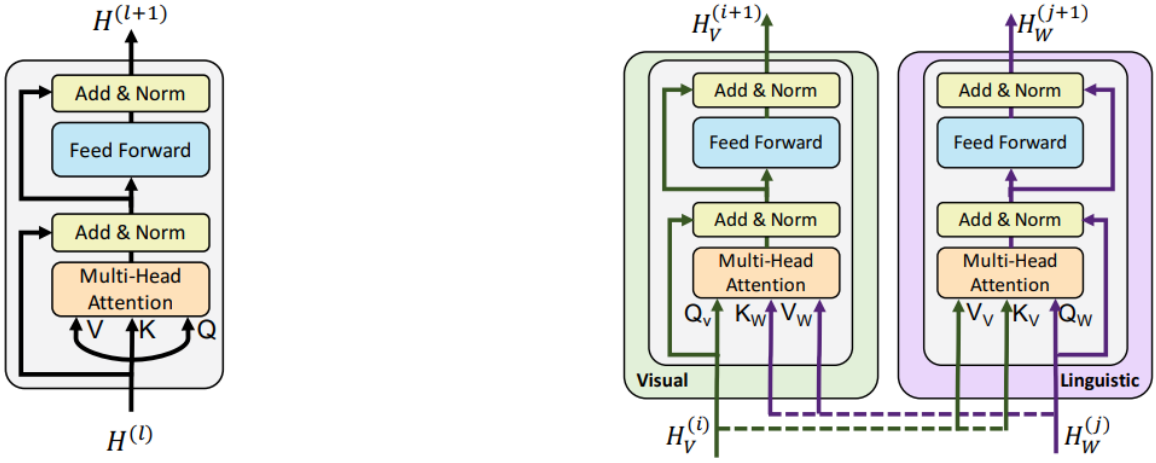

Figure 9: The co-attention mechanism of ViLBERT. [6]

- Multi-modal alignment prediction: the model must predict whether or not the caption describes the image content.

ViLBERT can be fine-tuned for many other tasks such as Visual Question Answering [12] and Caption-Based Image Retrieval [13]. This requires adding and training a task-specific classifier or regressor.

3.1.4 VisDial-BERT

ViLBERT [6] has been adapted to Visual Dialog [14] by modifying the input representation to accept longer sequence (10-round long conversation). First, the model is pre-trained on English Wikipedia and BookCorpus with the masked language modeling and next sentence prediction. Next, it is trained on the Conceptual Captions and VQA with the masked image region. Finally, the model is fine-tuned on sparse annotation by getting an image, a caption, a dialog history, a question and a list of 100 possible answers. The goal is to output a sorting of the answers.

3.1.5 VisualBERT

VisualBERT [15] is a model inspired by BERT. It allows processing text and images jointly and using the self-attention mechanism to align elements of the input text and regions of the input image.

VisualBERT is pre-trained on COCO image caption dataset (100k images with 5 captions each). The training contains 3 phases:

- Task-Agnostic
  - Some elements of text input are masked and must be predicted.
  - Given an image and two captions, decide whether the second one describes the image.
- Task-Specific: Train the model using the data of the task with the masked language modeling.
- Fine-Tuning by introducing task-specific input, output and objective.

3.1.6 VideoBERT

The main contribution of the VideoBERT [16] architecture is the possibility to learn high level video representations that capture meaningful and long-range structure. The model is pre-trained on YouTube cooking videos because spoken words are more likely to refer to visual content: 312K videos with a total duration of $23,186$ hours.

YouTube's automatic speech recognition (ASR) toolkit provided by the YouTube Data API is used to retrieve timestamped speech information. Videos are sample at 20 fps, and then a 30-frame clip is created. Finally, a ConvNet is applied to extract features and get a 1024-dimension feature vector.

3.1.7 VL-BERT

In VL-BERT [17], the visual feature embedding is newly introduced for capturing visual clues, while the other three embeddings follow the design of the original BERT paper. The visual geometry embedding is designed to inform VL-BERT the geometry location of each input visual element in the image. Each region of interest is then characterized by a 4-d vector denoting the coordinate of the top-left and bottom-right corner.

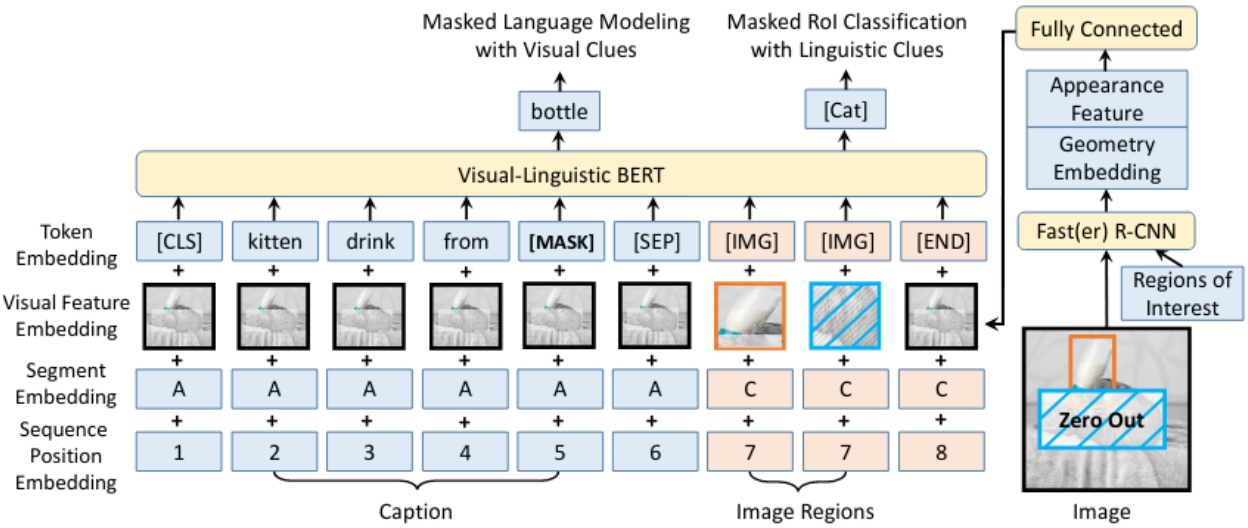

Figure 10: Architecture for pre-training VL-BERT. [17]

VL-BERT is pre-trained on both CC (captions are short) and BookCorpus+Wikipedia (text-only corpus to avoid over fitting on complex tasks).

- Task #1: Masked Language Modeling with Visual Clues
- Task #2: Masked RoI Classification with Linguistic Clues

Fine-tuning: the typical input formats Caption, Image and Question, Answer, Image.

In our experiments, we have used visio-linguistic models such as VilBERT in order to predict important moments. We describe the various approaches we have experimented with in the next section.

3.2 Predicting Media Memorability

The challenge we aim to tackle relates to the definition of what an important moment is and how to detect those in videos as for contributing to automatic story understanding. Concretely,

if we take the example of a user presented with a long video, we aim to provide him/her with a subset of the most important moments as we believe it might help him deciding whether he is interested in watching this content. Detecting highlights could also be especially relevant for very long and monotonous videos such as the ones taken with a Go Pro or a drone. In the context of TV broadcasts, a user could find interesting to automatically select the best TV moments that happened on TV for a particular day, just like the French TV program *Le Zapping*[4]. Achieving such a goal would hence enhance users data exploration experience by shedding lights into some content that would otherwise be lost in a sea of videos.

When it comes to formalising what an important moment is, [18] provides with a thorough overview of visual interestingness and related concepts. The authors argue than rather being a standalone concept, interestingness is closely linked to many aspects of subjective perception such as emotions, aesthetics or memorability and that there is a strong link between emotion and interestingness. More precisely, arousal has been found to be one of the most important attribute which explains interestingness [19]. The concept of memorability has also been explored. While several studies have concluded that it can be considered as "an intrinsic property of images" [20, 21], it is pointed out that it can been used to create video summaries [22].

3.2.1 Related Work on Important Moment Detection

The task of media *interestingness* [23] and media *saliency* [24] binary classification for video segments has gathered significant research attention. For this task, videos are segmented (generally into shots) and each segment manually labelled as being interesting or not.

Considering that key-shots have dependencies both with past and future frames in the sequence, [25] proposed a BiLSTM model. Adding attention layers also prove efficient [25, 26]. Recently, unsupervised models such as [27] using a reward functions for diversity and representativeness, also obtained results comparable with supervised models for visual video summarization. TVSum [28] and SumMe [29] presented in Table ?? are two datasets consistently used for this task. They both present the advantage of being annotated by more than 15 people. Indeed, the annotation of such videos is costly and such datasets are therefore rare and of small size. In this context, research on unsupervised models becomes particularly relevant as they will allow us to directly train on videos available throughout the MeMAD project. These MeMAD videos correspond to broadcaster Radio and TV programs that come from two content providers: Yle (*Yleisradio Oy*, Finland's national public broadcasting company) and INA (*Institut National de l'Audiovisuel*, a repository of all French radio and television audiovisual archives).

In addition to the information in Table ??, another characteristic about TVSum and SumMe is that audio was muted during the ground truth annotation process. Therefore, what is being said in the video is totally ignored when predicting important moments. We consider this fact to be a major limitation. Indeed, in real life, videos usually have sound and the assessment of whether a segment is interesting or not most probably also depend on non visual cues. For example, both dialog contents and other audio cues -is a person whispering? shouting?- are most likely relevant. Believing there is a gap to fill on the topic of multimodal moment extraction, we decided to consistently approach TV moments detection in a multimodal way and therefore to look for datasets annotated for audio-visual interestingness rather than using the two datasets aforementioned.

The second observation we are able to make from these two benchmark datasets is that the average length of the videos is quite low: 146 seconds for SumMe and 235 seconds for TvSum as shown in Table ??. We consider this property to be another limitation as in real life, it is more relevant to get a summary for a longer videos. Naturally, when video length increases so

[4] https://fr.wikipedia.org/wiki/Le_Zapping

Table 4: Datasets for video keyshots summarization [26]

Dataset	Videos	User annotations	Annotation type	Video length (sec)		
				Min	Max	Avg
SumMe	25	15-18	keyshots	32	324	146
TvSum	50	20	frame-level importance score	83	647	235

does the complexity of the story line. We therefore expect the summarization to become more difficult as well. We will therefore keep the high length of videos as a nice to have conditions for the datasets we will use.

Based on our readings on important moment detection, we formulate the following open questions, that we would like to contribute answering: Which modalities audio, visual, text is more useful in predicting video interestingness? How similar are the tasks of video memorability, saliency, interestingness prediction? Is it possible to build a model that would be reasonably robust to changes in datasets and importance proxies? Should long videos with a more complex story line be handled in a different way than short ones?

3.2.2 Approach 1: Combining Visual and Textual Features to Predict Media Memorability

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 at MediaEval[5] aims at fostering the research in the field by asking its participants to automatically predict both a short and long term memorability score for a given set of annotated videos. The full description for this task is provided in [30]. Last year's best approaches for both the long term [31] and short term tasks [32] indicated that high level representations extracted from deep convolutional models performed the best in terms of visual features. Furthermore, the best long term model [31] was a weighted average method including Bag-of-Words features extracted from the provided captions.

Following this approach, we (EURECOM and Aalto) created multimodal weighted average models with visual deep features and textual features extracted from both the provided video titles, as well as from automatically generated deep captions. In total, 23 teams did submit runs and our joint MeMAD submission got the best score for the "long-term" memorability subtask and the second best score of the "short-term" memorability subtask. Our implementation is available at
`https://github.com/MeMAD-project/media-memorability`. The full details of the approach is provided in Annex B.2. We summarize the key aspects in this section.

Visual Modality. **VisualScore.** Our visual-only memorability prediction scores are based on using a feed-forward neural network with visual features in the input, one hidden layer of 430 units and one unit in the output layer. We tested various combinations of hidden layer sizes and CNN-based visual features. The best performance was obtained with 6938-dimensional features consisting of the concatenation of I3D [33] video features, ResNet-152 and ResNet-101 [34] image features and two versions of SUN-397 [35] concept features. The image and concept features were extracted from the middle frames of the videos. The hidden layer uses ReLU activations and dropout during the training phase, while the output unit is sigmoidal. We trained separate models for the short and long term predictions with the Adam optimizer. The number of training epochs was selected with 10-fold cross-validation with 6000 training and 2000 testing samples.

[5] `http://www.multimediaeval.org/mediaeval2019/memorability/`

CaptionsA. Our first captioning model uses the DeepCaption software[6] and is quite similar to the best-performing model of the PicSOM Group of Aalto University's submissions in TRECVID 2018 VTT task [36]. The model was trained with COCO [37] and TGIF [38] datasets using the concatenation of ResNet-152 and ResNet-101 [34] features as the image encoding. The embed size of the LSTM network [39] was 256 and its hidden state size 512. The training used cross-entropy loss.

CaptionsB. Our second model has been trained on the TGIF [38] and MSR-VTT [40] datasets. First, 30 frames have been extracted for each video of these datasets. Then, these frames have been processed by a ResNet-152 [34] that had been pretrained on ImageNet-1000: we keep local features after the last convolutional layer of the ResNet-152 to obtain features maps of dimensions 7x7x2048. At that point, videos have been converted into 30x7x7x2048-dimensional tensors. A model based on the L-STAP method [41] has been trained on MSR-VTT and TGIF: all videos from TGIF, and training and testing videos from MSR-VTT have been used for training, and validation has been performed throughout training with the usual validation set of MSR-VTT, containing 497 videos. Cross-entropy has been used as the training loss function. The L-STAP method has been used to pool frame-level local embeddings together to obtain 7x7x1024-dimensional tensors: each video is eventually represented by 7x7 local embeddings of dimension 1024. These have been used to generate captions as in [41].

VisualEmbeddings. The local embeddings used for CaptionsB have also been used to derive global video embeddings, by averaging the mentioned 7x7 local feature embeddings. These global video embeddings have then been fed to a model of two hidden layers, the first one and the second one having respectively 100 and 50 units, and ReLU activation function. The number of training epochs is 200 with an early stopping monitor.

Textual Modality. Through initial experiments and from last year's results on this task, the descriptive titles provided with each video prove to be an important modality for predicting the memorability scores. In order to build on this observation, we generate captions for each video using the two visual models described above (**CaptionsA** and **CaptionsB**). While the generated captions are not always accurate, they seem to noticeably help the model disambiguate some titles and use some of the vocabulary already seen on the training set (e.g. the title contains words such as *couple"* or *"cat"* while the generated caption would say *"a man and a woman"* or *"an animal"*, respectively, which are more common words in the training set and thus help the model generalize better on inference time). The models described in this section use a concatenation of the original provided title and the generated captions as their input.

Multiple techniques for generating a numerical score from this input sequence were considered (in ascending order of their performance on cross-validation).

Recurrent Neural Network. We use an LSTM [39] to go through the GloVe embeddings [42] of the input and predict the scores at the last token. This model performed consistently the worst, probably due to the length of the input sequence at times, and the empirical observation that word order doesn't seem to matter for this task.

Convolutional Neural Network. We use the same model as [43] except for a regression head instead of a classifier trained on top of the CNN, and GloVe embeddings as input. This model leaks less information thanks to max-pooling, and performs much better than its recurrent counterpart.

Self-attention. Similar to the previous methods, we feed our input text to a self-attentive bi-LSTM [44] to generate a sentence embedding that we use to predict the memorability scores. This model performs on par with the CNN method.

[6] https://github.com/aalto-cbir/DeepCaption

BERT. We used a pre-trained BERT model [8] to generate a sentence embedding for the input by max-pooling the last hidden states and reducing their dimension through PCA (from 768 to 250). This model performs better than the previous ones but it is more computationally demanding.

Bag of Words. We vectorize the input string by counting the number of instances of each token (and frequent n-grams) after removing the stop words and the least frequent tokens. The score is predicted by training a linear model on the counts vector. This simple model performs the best on our cross-validation, which can be justified by the lack of linguistic or grammatical structure in the titles and generated captions that would justify the use of a more sophisticated model.

For all the models considered, the addition of the generated captions improves the prediction score on the validation set considerably. It also should be noted that the use of short-term scores for long-term evaluation yields substantially better results throughout all of our experiments.

3.2.3 Approach 2: Using Visio-Linguistic Models to Predict Media Memorability

In this second approach, we wanted to experiment with visio-linguistic models described in Section 3.1. We devise two variants adapted to the media memorability prediction task:

- The first variant is to freeze the pre-trained model weights and use them to infer and extract attention-pooled features for each modality (the output of the co-attention head represented in Figure 9). Those features (`pooled_output_t`, `pooled_output_v` and their fusion `pooled_output`) are used then to separately train a regressor to predict memorability scores.

- The second variant is to add a regressor on the top of the pre-trained ViLBERT and do an end-to-end training for the whole model.

All the code can be found in the ViLBERT forked repository at `https://github.com/bouzaien/vilbert-multi-task`.

Dataset Description. The dataset used is MediaEval 2018 containing a $8,000$-sample development set and a $2,000$-sample testing set. It contains the following fields:

- Video sources: videos are proposed in `.webm` format.

- Ground truth (only for the dev-set): video's name, short-term and long-term memorability scores, and number of annotations used to calculate scores.

- Pre-extracted visual features (e.g., HoG, Color Histogram, etc.)

- Additional data: LaMem dataset from MIT ($60,000$ images and their memorability scores).

Data Structure. As shown in the following directory tree, the ME data has nearly the same structure as existing datasets (e.g., VQA and NLVR2). This will help avoiding complex changes to the code in order to adapt it to this dataset.

```
|-- cache
|   |-- ME_test_23_cleaned.pkl
|   |-- ME_trainval_23_cleaned.pkl
|   |-- non_dc
|   `-- split
|-- features_100
|   |-- ME_test_resnext152_faster_rcnn_genome.lmdb
|   |-- ME_trainval_resnext152_faster_rcnn_genome.lmdb
|   `-- dc
|-- images
|   |-- dc
|   |-- test
|   `-- train
`-- out_features
    |-- train_dc_features.pkl
    |-- train_dc_features_nlvr2.pkl
    `-- train_features.pkl
```

The main sub-directories are:

- `cache`: it contains the cached `.pkl` files for the textual input.
- `images`: the frames extracted from videos are placed here.
- `features_100`: the visual features extracted from images.
- `out_features`: this sub-directory was added to save the image and text representations used to train a regressor.

Unlike fine-tuning for VQA and NLVR2, this task requires more data pre-processing in order to adapt it to the model input. These pre-processing tasks are detailed in this section.

After executing the ViLBERT end-to-end training script for MediaEval, the `vilbert_tasks.yml` config file will be used for the different training parameters, the loss function to be used and the dataset path. So, a ME task (ID 19) was added to this file as detailed in the listing 3.2.3. [ht] yamlvilbert$_{t}asks.yml$

Textual Input. The video captions are saved in the `dev-set_video-captions.txt` and `test-set_video-captions.txt` text files using video name-caption format for each video. The captions file looks like this:

```
video10.webm  couple-relaxing-on-picnic-crane-shot
video100.webm cute-black-and-white-cats-in-cage-...
video10000.webm owl-in-tree-close-up
```

In order to be able to feed the textual input to the model, it should have a specific format. To do so, the video IDs and the captions are extracted from the file, then we add scores, caption tokens and caption input mask. Finally, all list-type fields are tensorized to obtain the following format

```
{
    'video_id': 10,
    'caption': 'couple relaxing on picnic crane shot',
    'scores': tensor([0.9500, 0.9000]),
    'c_token': tensor([101, 3232, 19613, 2006, 12695, 11308, 2915, 102, ...,
    'c_input_mask': tensor([1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, ...,
    'c_segment_ids': tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
}
```

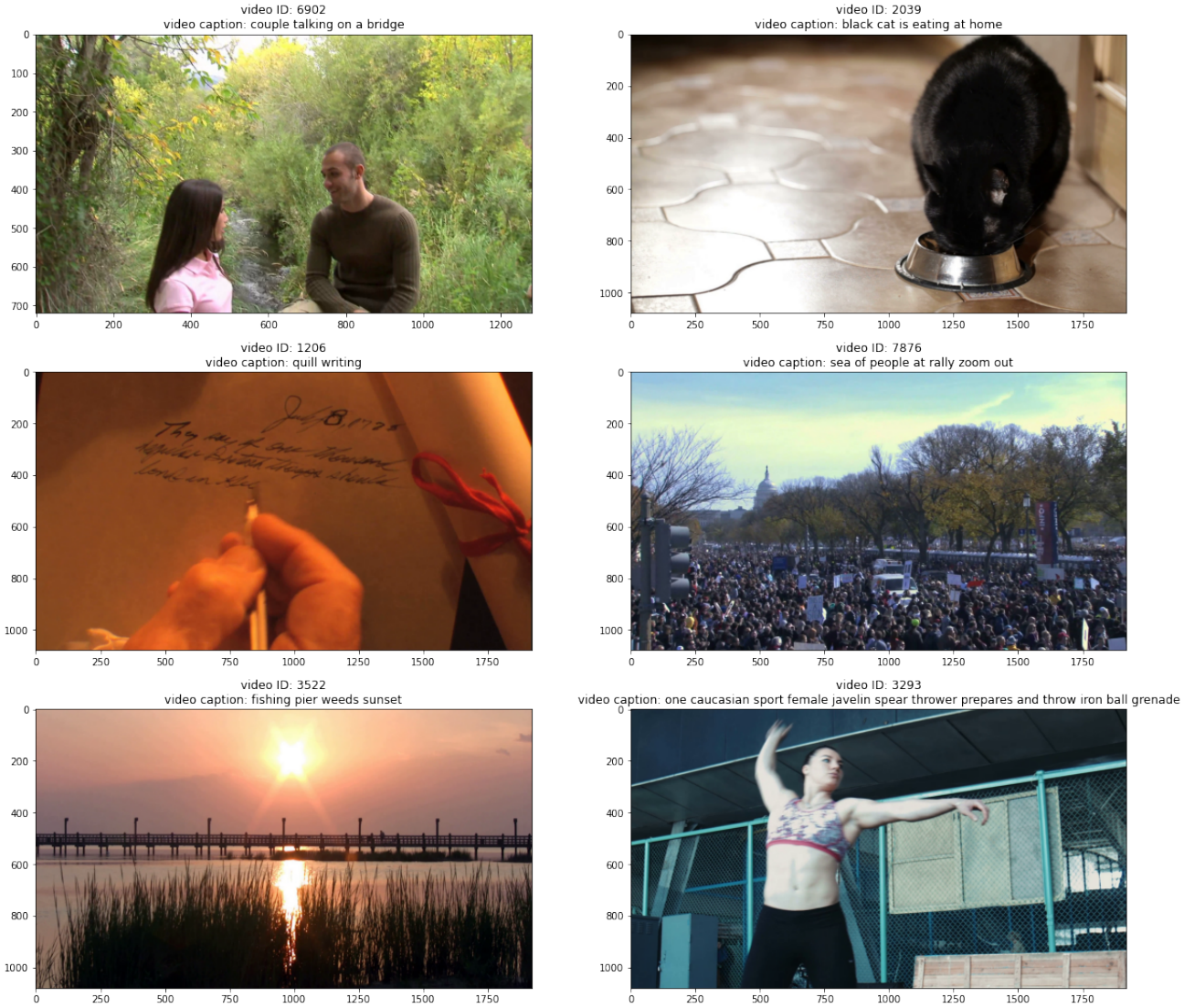

Figure 11: Extracted frames from random samples

Visual Input. Since we are dealing with short video streams, choosing the middle frame of each input can be a good trade-off between reducing the computations needed to treat the whole video and representing it without losing important features. As we can notice in Figure 11, there is a perfect match between the extracted frame and the descriptive caption.

The next step is to extract visual features from these frames. 100 feature boundary boxes are extracted for each representative frame using maskrcnn-benchmark [45]. The final visual input for each video is

```
{
    'bbox': numpy.ndarray([100, 4]),
    'num_boxes': 100,
    ...
    'image_id': str,
    'features': numpy.ndarray([100, 2048])
}
```

Another approach based on considering multiple frames was used later. It consists of randomly extracting 5 frames from each video, extract the visual features for each frame and then average them. As previously discussed, two variants are considered for the Media Memorability prediction task: Transfer Learning and Fine-Tuning. In both cases, a 4-layer neural network

with sigmoid activation function was added and trained to predict memorability scores. Also, many configurations were used such as using one and five extracted images, using the video captions and deep captions. Memorability task is evaluated using Spearman's rank correlation.

Variant #1: Transfer Learning. The first variant is transfer learning which consists of freezing the first layers of the pre-trained ViLBERT model and add a new trainable regressor on top of them. Finally, we train the new layers to predict video memorability scores.

Given the frozen weights, the whole dev-set ($8,000$ samples) is fed to the model. The output representations are saved on the disk to avoid redundant calculations while training different regressors. The output representations considered are textual H_W, visual H_V, their concatenation $[H_V, H_W]$ and their fusion (summation $H_V + H_W$ and multiplication $H_V * H_W$).

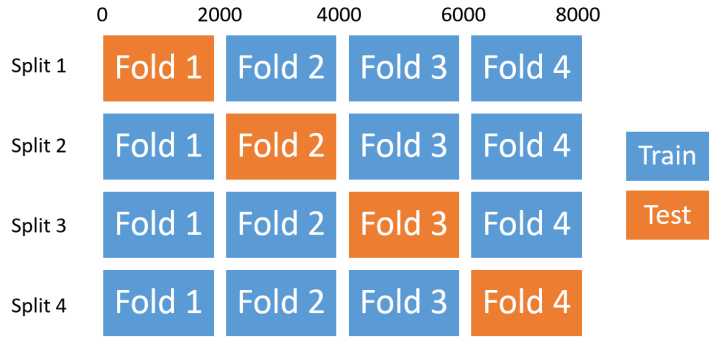

Figure 12: Cross-validation splits

Since the test-set was not available in the beginning, a 4-split cross validation was used on the dev-set to evaluate the performance of the models as shown in figure 12. Once the test-set was available, the regressor was trained on the whole dev-set and evaluated on the test-set. The different results of this approach are detailed in tables 5 and 6 for VQA and NLVR2 fine-tuned models respectively (full results are available here).

	H_V		$H_V * H_W$		$[H_V, H_W]$		$H_V + H_W$	
S1	0.439	0.217	0.437	0.211	0.440	0.210	0.447	0.216
S2	**0.466**	**0.237**	0.464	**0.247**	0.472	**0.234**	**0.476**	**0.242**
S3	0.457	0.217	0.458	0.237	0.458	0.211	0.468	0.222
S4	0.463	0.232	**0.466**	0.225	**0.477**	0.233	0.463	0.236
Mean	0.456	0.226	0.456	0.230	0.462	0.222	0.463	0.229
test-dev	0.453	0.238	0.446	0.231	0.455	0.247	0.459	0.248

Table 5: Transfer learning results based on the VQA fine-tuned model

	H_V		$H_V * H_W$		$[H_V, H_W]$		$H_V + H_W$	
S1	0.412	0.209	0.421	0.205	0.422	0.219	0.422	0.215
S2	0.449	0.219	0.462	**0.244**	0.463	0.226	0.463	0.233
S3	0.442	0.237	0.457	0.242	0.443	0.239	0.455	0.243
S4	**0.462**	**0.242**	**0.475**	0.240	**0.484**	**0.253**	**0.470**	**0.246**
Mean	0.441	0.227	0.454	0.233	0.453	0.234	0.453	0.234
test-dev	0.417	0.208	0.427	0.215	0.416	0.209	0.428	0.217

Table 6: Transfer learning results based on the NLVR2 fine-tuned model.

Variant #2: Fine-tuning. This approach consists of unfreezing the pre-trained models and re-train them on the MediaEval data after adding the task-specific regression layers. Only the multiplication fusion between textual and visual representation is used in this part considering the results of the previous approach.

Given the huge number of trainable parameters of the model and the reduced size of the dataset, training for 20 epochs resulted in over-fitting problems as shown in figures 13, 14 and 15. For different model, different numbers of training epochs are considered to avoid this problem (Table 7).

Model	Epochs	STM	LTM
Base	13	0.490	**0.254**
VQA	15	0.486	0.227
NLVR2	12	**0.491**	0.228

Table 7: MediaEval fine-tuning results

3.2.4 Results and Analysis

During the evaluation process, we created four test folds of 2000 videos and therefore four models trained on 6000 videos. For the VisualScore approach, we decided to use predictions from a model trained on the entire set of 8000 videos (VisualScore8k), as well as the mean predictions from the combinations of the four models trained on 6000 videos (VisualScore6k). For the Long Term task, all models except from the WA3lt exclusively use short-term scores.

- WA1 = 0.5Textual+0.5VisualScore
- WA2 = 0.25Textual+0.25VisualEmb+0.5VisualScore8k
- WA3 = 0.25Textual+0.25VisualEmb+0.5VisualScore6k
- WA3lt = WA3 with long-term scores

We observe that the weighted average method (Approach 1) which was trained on the whole training set and included our two visual approaches and our textual approach works the best for short term predictions. For long term prediction, one of the key observations to make is that WA3lt got the second worst results. This is consistent with our early observation that short-term scores for long-term evaluation yields substantially better results.

We also observe that:

- The best model is usually not the same for both scores (Table 7).
- There is no significant improvements when selecting multiple frames (see full results at `https://docs.google.com/spreadsheets/d/1XqRj6egkbiFk-hJYby-Qu0EQdT7I3uAbQ1067UTofIw/edit?usp=sharing`).

Table 8: Results on test set for short term memorability

Method	Spearman	Pearson	MSE
Textual	0.441	0.464	0.01
VisualScore	0.495	0.543	0
WA1	0.512	0.552	0
WA2	**0.522**	**0.559**	0
WA3	0.520	0.557	0

Table 9: Results on test set for long term memorability

Method	Spearman	Pearson	MSE
Textual	0.239	0.25	0.03
VisualScore	0.268	0.289	0.03
WA2	**0.277**	**0.296**	0.03
WA3	0.275	0.295	0.03
WA3lt	0.260	0.285	0.02

- Using the representation fusion or concatenation has slightly improved the results.
- End-to-end training can cause the model to over-fit because of the reduced size of the dataset compared to the number of model parameters.

Team	Best STM Score	Best LTM Score
Insight@DCU	**0.528**	0.270
MeMAD	0.522	**0.277**
ViLBERT	0.491	0.254
UPB-L2S	0.477	0.232
RUC	0.472	0.216
EssexHubTV	0.467	0.203
TCNJ-CS	0.455	0.218
HCMUS	0.445	0.208
GIBIS	0.438	0.199

Table 10: ViLBERT study and the MediaEval 2019 results.

In conclusion, while the use of visio-linguistic model is a promising research avenue (Approach 2), we still observe that our simpler multimodal weighted average method (Approach 1) provides the best results for the Predicting Media Memorability Task. One of the key contribution of this approach is to have demonstrated that using deep captions helped improving the predictions. We also conclude that, quite surprisingly, a simple n-gram frequency count was more efficient at modelling memorability than more sophisticated textual models. Finally, the fact that long term memorability was better predicted using short term predictions indicates that we failed at capturing the memorability decay of a scene from a few minutes to a few days. In the future, we would like to focus more on this aspect of the task.

3.3 Predicting Media Interestingness

3.3.1 Motivation

Building up on the success of our participation to the Mediaeval Memorability task, some further research has been conducted on the topic of important moment extraction. With the memorability task we have learned that individuals with different interests and background tend to remember and forget the same video clips. Furthermore, despite having no information on these individuals and their sensibilities, we were able to automatically infer a memorability score to audio-visual segments solely based on their content.

Are there other properties, which could at first be considered as purely subjective and individual, on which individuals agree and that could be automatically predicted? A literature review taught us that the tasks of interestingness binary classification for video segments has gathered significant research attention. However, in the two datasets consistently used for this task: TVSum [28]and SumMe[29], audio has been muted. Therefore what is being said in the video is totally ignored when predicting important moments. This was also the case for the

Memorability MediaEval task. We consider this to be a major limitation. Indeed, in real life videos usually have sound and the assessment of whether a segment is interesting or not most probably also depends on non visual cues. For example, both dialog contents and other audio cues -is a person whispering? shouting?- are most likely relevant. Believing there is a gap to fill on the topic of multimodal moment extraction , we decided to consistently approach TV moments detection in a multimodal way.

The second observation we were able to make from these two benchmark datasets is that the average length of the videos was quite low : 146 seconds for SumMe and 235 seconds for TvSum. We consider this properties to be another limitation as in real life once again, it is more relevant to get a summary for a long video rather than for a video which is already short. Naturally, when video length increases so does the complexity of the story line. We therefore expect the summarization to become more difficult as well.

The questions we would like to answer can be formulated as follows : Which modalities audio, visual, text is more useful in predicting video interestingness? How similar are the tasks of video memorability, saliency, interestingness prediction? Is it possible to build a model that would be reasonably robust to changes in datasets and importance proxies? Should long videos with a more complex story line be handled in a different way than short ones?

3.3.2 Model Architecture

We start by presenting the models we worked with. As explained in the previous section, multimodality is the first angle chosen to tackle the task of moment extraction. Consequently, the first question we need to answer is : how to obtain multimodal inputs from a video? Similarly as for the Memorabililty task, the visual vectors we use as inputs for our model consist in the concatenation of I3D [33] video features, ResNet-152 and ResNet-101 [34] image features and two versions of SUN-397 [35] concept features. We call them the Picsom features. Semantics being most easily expressed in words, we believed using Automatic Speech Recognition (ASR) would allow our model to benefit from the huge work carried out in the field of Natural Language Processing on the topic of semantics. We then generated subtitles from the video audio input. Finally, following the success of the Memorability task approach, we also generated deep captions. We experiment with three ways of transforming text to feature vectors : TF-IDF, Word2vec [46]and BERT [47]. For the Memorability Task, we treated the text and the visual modalities independently and built a distinct feed-forward neural network for each modality. We now experiment as well with a model that fuses feature vectors of the different modalities. Here we take inspiration from the best visual interestingness classification approaches. There has been some work done with bidirectional longshort-term memory (BiLSTM)[25] as keyshots have depencies both with past and future frames in the video sequence. The models can also include attention layers [25] [26]. Willing to stay close to this type of approaches while introducing multimodality, we explored models from multimodal video sentiment analysis. We found [48], one of the top performing models (on the Mosi dataset [49]) which is also a BiLSTM with an attention layer between video segments but also with an attention layer between modalities, and decided to experiment with this model.

This model can also be used in a unimodal fashion if we only want to experiment with a unimodal BiLSTM model. Given the sequential nature of videos, such a model is likely to be relevant. We first replicated their study with the MOSI sentiment analysis dataset and the extracted features provided by the authors and obtained results comparable with the authors (Table 11).

Table 11

Modality	Paper's result	Replication
Text(T)	55.8	56.8
Visual(V)	78.1	80.2
T+V	80.2	78.5

3.3.3 Experiments on the Cognimuse database

The first database we found that matched our requirements - videos longer than 1-2 minutes with audio included- was Cognimuse [50]. It contains, among others, long videos of minimum 15 minutes annotated with a ground truth for visual saliency, audio saliency and audiovisual saliency. The binary annotations are available at the frame level and were produced by three annotators in separate runs for each saliency layer. The annotation interface used by Cognimuse as well as some frame examples can be seen in Figure 17 from [50]. We first tried to replicate the results obtained in [51]for the visual modality by using a Keras implementation [7] of the Caffe model they used and using the parameters described in their paper. Each video was this splited in clips of 16 frames and one vector of C3D feature was extracted per sequence. The last sequence with less than 16 frames was padded. We obtained a score of 0.65 for AUC (Area Under the Curve) when the authors obtain 0.72 for the task of visual saliency. The authors informed us that they are planning to release the code for this paper as well as for a more recent one [52] but were delayed because of Covid-19. Given that the results we obtained with the textual modality were also not encouraging (the best score obtained was 0.52 for Bert features fed to a Feed Forward network), we decided to keep our experiments with this database are on hold until the authors release their code.

3.3.4 Experiments on the MediaEval 2017 dataset

The next dataset we considered is the Media Interestingness Dataset [23]. It contains Interestingness binary annotations for 103 Hollywood like movies trailers and 4 continuous extracts of ca. 15 min from full- length movies. Sound is included and no text is provided. The results are expressed in terms of Mean Average Performance (MAP). The best competitors on this dataset obtained a MAP of 0.212. With the feed-forward model, we obtained an MAP of 0.115 with the he C3D features provided by the organisers. For the textual modality, made of generated deep captions, we obtained 0.120 using TF-IDF. Given that we obtained similar results when Picsom features, Word2Vec and BERT, we here only present the results for TF-IDF and C3D. After production of the ASR, we realised only 10 percents of the video segments are associated with text. This is most likely due to the the nature of the dataset that mostly contains trailers. In order to be able to compare ASR features performance with other features, we also experiment with the subset that only contains segments for which an ASR segments was available. Table 12 and 13 summarize the results obtained with the BiLSTM model using C3D for visual features and TF-IDF for textual features, for the entire dataset and the ASR subset.

The first observation we can make is that regardless of the approach neither feed-forward model (state of the art method for Memorability prediction) nor the BiLSTM with attention produced results reached the scores of the best approaches for this task. This could suggest that the models performing well for Memorability prediction Multimodal sentiment analysis do not generalise to Interestingness Prediction . It could also mean that the approaches tested are not very robust across datasets and types of videos. We need to precise that for text we used generated deep captions or ASR when for the sentiment analysis task, the transcript

[7] https://gist.github.com/albertomontesg/d8b21a179c1e6cca0480ebdf292c34d2

Table 12: Mean Average Precision(MAP) for the entire dataset

Method	MAP
Text(T)	0.117
Visual(V)	0.125
T+V score with attention	0.127
T+V score without attention	0.122

Table 13: Mean Average Precision(MAP) for the ASR subset

Method	MAP
TextCaptions	0.144
TextASR	0.146
Visual	0.144
TextCaptions+VscoreWithAttention	0.137
TextASR+VScoreWithAttention	0.136
TextCaptions+VScoreWithoutAttention	0.147
TextASR+VScoreWithoutAttention	0.147

was available. This could account for some differences with regards to the text modality performance. However, we still observe that for the subset, ASR text slightly outperforms the visual modality and that the performance of the model does not increase much when adding visual features. The results are consistently worst when adding the attention layers. Another general observation is that results obtained for the subset are better across modalities. A potential explanation is that the subset is less imbalanced than the general dataset . It has a ratio of 7 uninteresting segments for one interesting segments. when this ratio is 25 for the whole dataset. Even if we do take into account the imbalanced classes problem by adding the corresponding weight to the underrepresented class, it could still play a role.
To conclude, despite not reaching the scores obtained by the best approaches on this task, we investigated new approaches and have a usable model for multimodal sentiment analysis. The best approach on the MediaEval2017 Interestingness task was obtained using movie genre [53]as an intermediate representation. Perhaps other intermediate representations such as sentiments or dialogue acts [54], able to generalise to other datasets, could be investigated. Finally, as recently Transformers have been performing well on multiple tasks , we are also currently working on adapting Vision-and-Language BERT [6] to videos and are planning on experimenting with it.

4 Moments Enrichment

During this second year period, we continue to investigate the role of named entity recognition and disambiguation as a way of enriching TV program segments with background knowledge, typically coming from encyclopedic knowledge base such as Wikidata.

The MeMAD consortium partners have tackled both the multilingual challenge (being able to process content in English, French, Finnish and Swedish) as well as the challenge of analyzing texts that are the results of an automatic speech recognition process and that can thus be grammatically incorrect. In the remaining of this chapter, we first describe a NER system developed by Aalto to work on ASR transcripts (Section 4.1.1). Second, we describe the Lingsoft NER system which is based on rules (Section 4.1.3). Both systems have been evaluated on the standard FiNER dataset so that the approaches can be compared. Third, we present the EURECOM approach to extract fine grained entity types on the standard TAC KBP benchmark (Section 4.4). Finally, we describe the EURECOM NER experiments on the French MeMAD datasets (Section 4.5) as well as an experiment in extracting named entities on French ASR (Section 4.6).

4.1 NER methodologies

4.1.1 Aalto NER

Aalto has studied a deep neural network (DNN) approach for fully data driven learning in named entity recognition (NER) on low-resourced languages [55]. The system was first benchmarked with other approaches for Finnish texts in manually labelled FiNER corpora (DigiToday and Wikidata). As the results were really good we continued into experiments with our target task which is NER on transcripts provided by ASR.

NER on ASR transcripts is motivated by the goals of MeMAD where we aim to enrich the broadcast material with content descriptions and linking. Most often the available metadata is very limited, so the descriptions are mainly based on ASR transcripts. Applying NER on noisy data which contains ASR errors such as inserted, deleted and substituted words and excludes punctuation, capitalization and lacks a proper sentence structure has not been explored sufficiently before.

The NER on ASR experiments included one part where our ASR performs well and corresponding correct transcripts are available for comparisons (Parliament sessions) and another more demanding part for ASR (Pressiklubi talk show). As no manually checked named entities are available for ground truth, the experiments are limited to comparisons to a rule-based baseline method (Lingsoft NER) and NER on manual transcripts where they are available.

In order to achieve competitive results on noisy low-resource data we implemented a conventional conditional random field (CRF) system as a layer on top of a bidirectional long short-term memory (BiLSTM) DNN architecture that utilizes words, characters and morph units. This way we avoid the need for handcrafted features and word embeddings that would require big training data. Furthermore, the combination of words, characters and morphs has previously provided low out-of-vocabulary rates and the best language models for morphologically rich languages, such as Finnish.

The methods to deal with the low-resource setting included the use of transfer learning from language resources such as named entity lists and embeddings in other related languages, for example Estonian. We also improved the robustness for noisy data, by transforming the training data to resemble more to ASR transcripts by removing punctuation and capitalization.

4.1.2 Methods

The architecture that we used for the experiments is a BiLSTM with a CRF layer on top. The architecture utilizes words, characters and morphs. We used the pre-trained fastText word embeddings [56]. The character and morph embedding were learned from scratch during the training of the network. When the network learns to predict the classes, it also learns those embeddings. So, basically the same datasets that were used for training the NER system were also used for learning the embeddings. Morphs were obtained using the Morfessor toolkit [57]. The input for the system is a whole sentence and the output are the named entities found in that sentence. The architecture is presented in Figure 18.

In order to improve the performance in different domains, we used multilingual embeddings aligned in a single vector space provided by MUSE [58]. As a source language, we used Estonian language because it has similar morphological and sentence structure to Finnish. We used those embeddings to do a nearest neighbor search from Estonian to Finnish language and used that as a direct translation from source to the target language. The tags are then directly transferred from Estonian to Finnish.

Because some of the translations were not very accurate, we used thresholding and kept only the translations that have high nearest neighbor candidate score in the target language. We did multiple experiments and found that a threshold value of 0.6 yields best results.

Since personal names and location names are almost the same in Finnish and Estonian, we kept them as they are in the Estonian and just transferred them to Finnish. This approach gave us an improvement over translating them as we did with the other entities.

4.1.3 Lingsoft Analyzer with semantic linking

Lingsoft and LLS have provided a NER service with semantic linking for the consortium through an API. The NER is based on the Lingsoft proprietary language analyser technology, which in turn is built on two-level morphological analyzer [59], finite-state transducers and constraint grammar [60] disambiguation to decide on the correct reading in a given context. The NER service also supports multi-word expressions. In addition, heuristic rules are used for the named entity recognition (Figure 19). The analyzer has approximately 250 rules per language. As resources, the analyzer requires

- an analyzer lexicon for morphology;
- Constraint Grammar (CG) rules;
- Heuristic NER rules;
- and a semantic layer as a form of semantic lexicons built from ontologies.

The NER service is available through an API. Both plain text and various timed text versions such as Lingsoft proprietary json, Limecraft Flow's proprietary format, and standard formats such as SRT are supported. In the following, a sample call and a sample result using Limecraft Flow timed text format are presented as an example.

```
{
  "language":"fi",
  "domain":"NER+Wikidata_memad_flow",
  "text": { "options": {}, "words":
  [
    {"word":"Tassä ", "start":0.0, "end":1.0 },
    {"word":"on ", "start":1.1, "end":2.0 },
    {"word":"Sauli ", "start":2.1, "end":3.0 },
```

```
    {"word":"Niinistö", "start":3.1, "end":4.0 },
    {"word":".", "start":4.1, "end":5.0 }
  ]
  },
```

Listing 5: JSON-formatted call for the NER API with timed text


```
{
   "KeywordPersonNames": [
      {
         "URI": "Sauli_Niinistö/person_names",
         "keyword": "Sauli Niinistö",
         "source": "person_names",
         "relevance": 4.605170185988092,
         "frequency": 1,
         "broader": [],
         "path": [
            [
               "ner",
               "person names"
            ]
         ],
         "contexts": [
            {
               "timeCodes": {
                  "contextBegin": 1.1,
                  "contextEnd": 5.0,
                  "begin": 2.1,
                  "end": 4.0
               },
               "context": [
                  "Tässä on ",
                  "Sauli Niinistö",
                  "."
               ]
            }
         ]
      }
   ]
}
```


Listing 6: JSON-formatted call result from the NER API for timed text

The NER has been improved in late 2019 and early 2020 by including more current names in the lexicon both in Finnish and Swedish. In addition, the NER output categories have been standardized to include typical NER categories available in FiNER[8]: (PERSON, ORGANIZATION, PRODUCT, LOCATION, DATE and EVENT), which were previously partially not covered by the Lingsoft service.

In the NER update of 2019, three new NE categories were added: products (PRO), which contains 1) known product names defined in the NER lexicon and 2) heuristically detected product names, for example ”Samsung Galaxy S9”, ”Toyota Yaris”, ”Hiljaisii Heeroksii - kappale” (the song ”Hiljaisii Heeroksii”); organizations (ORG), which contain 1) own organization names defined in the NER lexicon and 2) heuristically detected organization names such as ”NATO”, ”Suomen Akatemia” (Academy of Finland) or ”Tampereen kirjallisuusyhdistys” (the Literature society of Tampere); and events (EVENT), which contains 1) known event names defined in the NER lexicon and 2) heuristically detected event names, such as ”Slush”,

[8] https://github.com/mpsilfve/finer-data

”Flow-festivaali” (the Flow Festival), ”Ukrainan sota” (Ukranian war). The complete list of the NER types and their correspondence to the IOB types is given in the table 14

Entity	Corresponding Lingsoft Entity
PERSON	Persons
LOCATION	Place names
ORGANIZATION	Company names, Organizations
PRODUCT	Product
EVENT	Event
DATE	Date, Year numbers
OTHER	Business ids, Email address, IBAN account numbers
	ISBN numbers, Medicine names, Origins
	Person IDs, Phone numbers
	Registration plates, Street addresses, Times

Table 14: List of Lingsoft NER entities and correspondence to IOB Entities

In addition, context hints are used to classify results into more specific categories, instead of simply labeling them as to ”unclassified names”. This is due to more context-aware heuristics and utilizing detected context hints also in other contexts, where no hints are present. E.g., if the text explicitly mentions XYZ as a company, this knowledge can be used to tag it as a company name in both cases in: ”**Teknologiayritys** XYZ rekrytoi ihmisiä. [...] Uutisissa mainittiin XYZ.”[9]

Semantic linking is carried out by querying Wikidata with those instances (words or multi-word expressions) that Lingsoft NER has recognized. The Wikidata hits are further disambiguated based on ’instance of’ -category of Wikidata. Thus those hits that seem improbable based on the categories recognized in the full article are weeded out. A result for a Wikidata call is given in the following.


```
"KeywordWikidata": [
      {
          "URI": "http://www.wikidata.org/entity/Q29207",
          "keyword": "Sauli Niinistö",
          "source": "wikidata",
          "relevance": 4.605170185988092,
          "frequency": 1,
          "broader": [
              "ihminen"
          ],
          "path": [
              [
                  "wikidata",
                  "ihminen"
              ]
          ],
          "contexts": [
              {
                  "timeCodes": {
                      "contextBegin": 1.1,
                      "contextEnd": 5.0,
                      "begin": 2.1,
                      "end": 4.0
                  },
                  "context": [
                      "Tässä on ",
```


[9]”Technology **company** XYZ is recruiting more personnel. [...] XYZ was mentioned in the news.”

```
                    "Sauli Niinistö",
                    "."
                ]
            }
        ]
    }
  ]
}
```


Listing 7: JSON-formatted call result from the NER API Wikidata query for timed text

Additionally, the usefulness of the Wikidata links and the corresponding multilingual labels obtained this way is possibly tested in the second round of evaluation especially for multilingual search. In this case, the multilingual Wikidata labels can be used to find information in material in a language the user does not know.

4.2 Finnish NER benchmarking results

4.2.1 Finnish FiNER test set

To benchmark the Finnish Named Entity Recognition models developed in the MeMAD project, we first use the FiNER corpus, for which the state-of-the-art Finnish results have been published. The FiNER corpus [61] consists of a texts collected from an online technological news service DigiToday in Finnish.[10] The dataset contains 6 types of named entities. They are Person (PER), Location (LOC), Organization (ORG), Product (PROD), Event (EVENT) and Date (DATE). The class distribution for this dataset is shown in Table 15. In addition to the in-domain Digitoday test set, the FiNER corpus also contains an out-of-domain Wikipedia test set. Table 16 shows the class distribution of the Wikipedia test set.

Table 15: The class distribution in Digitoday dataset based on manual labeling

Class	Count
ORG	15445
LOC	4159
PER	6517
DATE	3685
PRO	11655
EVENT	569
TOTAL	42030

Table 16: The class distribution in Wikipedia test set based on manual labeling

Class	Count
ORG	1821
LOC	1427
PER	2492
DATE	1862
PRO	2135
EVENT	362
TOTAL	10099

As an evaluation metric, we used the micro-average F1 score. Micro-average will aggregate the contributions of all classes to compute the average metric, which is a preferred approach when dealing with imbalanced classes. In Tables 17 and 18, we report the benchmark results

[10] It is a public dataset which can be obtained from `https://github.com/mpsilfve/finer-data`.

for this data set for two MeMAD partner NER models, the Lingsoft's rule-based NER and the Aalto NER developed during the project. We compare our models to the original rule-based FiNER NER [61] [11] and a new neural-network based NER model based on Finnish version of BERT (FinBERT NER) [62], which are current state-of-the-art NER models for Finnish. It is important to note that the Finnish Named Entity Recognition state-of-the-art results F1 scores cannot be directly compared to those in other languages, due to the fact that the language resources to build such models are fairly scarce and Finnish is a highly inflecting language, in which the names also inflect, which makes the NER task more difficult.

The authors of the FinBERT NER only report the micro averages, hence no other measures were available for comparison. In both of these evaluations, the FinBERT NER trained with the Digitoday training set beats the others. The precision of the other models is at the same level, but the all the other models suffer from poorer recall.

Table 17: Comparison of Finnish NER models for in-domain Digitoday test set. The FiNER results are from [61] and the FinBERT NER results are from [62].

	Aalto			Lingsoft			FiNER			FinBERT NER		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
DATE	96.04	98.64	97.33	95.79	91.91	93.81	97.92	98.74	98.33	-	-	-
EVENT	51.85	48.28	50	82.14	79.31	80.7	100	100	100	-	-	-
LOC	95.41	85.66	90.27	93.81	92.13	92.96	92.53	94.52	93.51	-	-	-
ORG	85.91	90.84	88.31	92.27	79.16	85.21	93.48	85.57	89.35	-	-	-
PER	76.15	90.21	82.59	85.45	89.57	87.46	87.76	83	85.32	-	-	-
PRO	84.32	75.65	79.75	83.27	69.61	75.83	82.49	71.18	76.41	-	-	-
Micro Avg	85.5	85.35	85.42	88.83	78.88	83.56	90.41	83.51	86.82	91.30	93.52	92.40

Table 18: Comparison of Finnish NER models for Out-domain Wikipedia test set. The FiNER results are from [61] and the FinBERT NER results are from [62].

	Aalto			Lingsoft			FiNER			FinBERT NER		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
DATE	95.21	98.16	96.66	96.31	92.59	94.41	97.3	96.52	96.91	-	-	-
EVENT	40.55	24.51	30.56	75.84	36.49	49.27	70.87	57.69	63.6	-	-	-
LOC	85.75	71.8	78.16	81.86	72.18	76.72	83.88	77.71	80.67	-	-	-
ORG	62.7	70.46	66.35	72.62	39.85	51.46	79.3	51.28	62.28	-	-	-
PER	90.52	88.35	89.42	86.61	72.54	78.96	85.32	77.79	81.38	-	-	-
PRO	75.47	73.72	74.59	68.14	18.96	29.66	73.8	49.32	59.12	-	-	-
Micro Avg	80.84	79.22	80.02	84.33	57.51	68.39	85.17	72.47	78.31	80.61	82.35	81.47

The results show that the Aalto system developed in MeMAD has reached closer to the Finnish NER state of the art, with Lingsoft NER a few percentage points behind for the in-domain test set, and bit more for the out-domain test set. The Lingsoft rule-based NER suffers from poorer recall, especially in out-domain Event and Product categories, which are probably more difficult to catch with a rule-based system. What the Lingsoft NER service has, though, is a product-quality reliable and fast text analysis pipeline and an API with several input and output formats in addition to plain text, such as TSV, CSV, subtitling formats such as SRT and WebVTT, a possibility to link to external ontologies such as Wikidata and the General Finnish Ontology. In addition, the FinBERT NER and FiNER models are experimentally hosted by Lingsoft and can be used through the Lingsoft API. Including Aalto model via the Lingsoft API

[11] Available online at https://github.com/Traubert/FiNer-rules

is underway. The results from these external NER models can also be to certain extent normalized (lemmatized) via the Lingsoft language analysis tools and further linked to external ontologies similarly to what is possible with the Lingsoft's proprietary NER.

4.2.2 Extending Yle MeMAD broadcast media evaluation set for NER in Finnish and Swedish

Until now, there has been little evaluation data for named entities in the MeMAD domain for Finnish and Swedish. In 2020 Yle, Lingsoft and LLS worked together to extend the Yle MeMAD broadcast data evaluation set to additionally include NER annotations. The data set was originally developed to evaluate the quality of the speech recognition and diarisation, reported in D2.1 and D2.2.

The programs contain a variety of topics. In the Finnish set, the topics range from a magazine program devoted to consumer issues in which e-bikes and driving school are discussed, to talk show type current events discussion programs and European Parliament election debates. In the Swedish set, there are also current events talk shows, European Parliament election debates, and a craft and cooking show, each with a variety of different topics.

The annotations were manually created following the same annotation conventions as the FiNER set[61], excluding nested labels. In addition to the transcripts, we extended the data set with hard-of-hearing subtitles for the same programs, and carried out annotations for the subtitle set.

The length of the Finnish test set is approximately 5 hours of video material (7 different program items) and there are 1345 sentences in the transcription test set in total. The length of the Swedish test set is approximately 5.5 hours of video material (9 different program items) and there are 1466 sentences in the test set. In the Finnish subtitle set there are 3086 subtitle blocks in total. In the Swedish subtitle set, there are 3850 subtitle blocks in total. The number of subtitle blocks is considerably higher than the number of sentences in the transcriptions, due to the space constraints in subtitling: each subtitle block can only contain maximum of two rows and approximately 40 characters per row, and thus sentences are often split into several subtitle blocks. The class distributions for the Finnish and Swedish datasets are given in Table ??.

Table 19: The class distribution in Finnish and Swedish Yle test set based on manual labeling

	Finnish		Swedish	
Class	Subtitles	Transcripts	Subtitles	Transcripts
ORG	511	442	247	296
LOC	478	545	357	464
PER	566	637	383	563
DATE	12	6	32	50
PRO	71	30	127	168
EVENT	9	6	21	24
TOTAL	1647	1666	1167	1565

4.2.3 Benchmarking results with the Yle MeMAD broadcast evaluation set for NER

We tested all four NER models also with the Finnish Yle MeMAD NER test set. Contrary to the previous evaluation, in which the results for FinBERT NER are from [62], the comparison results reported here for FinBERT NER and FiNER are from a Lingsoft-hosted installations. These results are reported in Tables 20 and 21. As we can see, all four models perform at comparable level, with FinBERT NER again being slightly better than the others. The runtime comparison (evaluation of the entire data set) is presented in Table 22. It clearly shows that

while the performance of the FiNER model is slightly better than both Aalto and Lingsoft models, the wait time is very long, and thus it is not suitable for cases in which the queries need to be returned fast. The models developed in MeMAD (Aalto and Lingsoft) are easily fastest, with the performance of the FinBERT NER also in a reasonable range. The machine learning models of Aalto and FinBERT NER have an additional initialization time for loading the model and embeddings into the memory, but if the server is already up and running, it will not affect the subsequent queries. Hence the initialization time is reported separately.

Table 20: The comparison on the NER performance on the subtitle set of the Finnish Yle test data

	Aalto			Lingsoft			FiNER			FinBERT NER		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
DATE	64.71	91.67	75.86	45.45	83.33	58.82	62.50	83.33	71.43	78.57	91.67	84.62
EVENT	40	22.22	28.57	25.00	22.22	23.53	25.00	22.22	23.53	0	0	0
LOC	93.53	94.12	93.82	91.46	94.54	92.98	93.13	93.91	93.51	91.39	93.70	92.53
ORG	73.81	67.52	70.53	84.76	62.04	71.64	88.37	66.93	76.17	86.82	68.30	76.45
PER	91.21	94.3	92.73	96.49	92.23	94.31	95.53	86.93	91.03	96.73	88.87	92.63
PRO	34.38	15.71	21.57	37.50	12.68	18.95	45.00	12.68	19.78	58.21	54.93	56.52
MICRO AVG	85.21	82.18	83.67	89.66	79.64	84.35	91.30	79.15	84.79	90.41	81.95	85.97

Table 21: The comparison on the NER performance on the transcription set of the Finnish Yle test data

	Aalto			Lingsoft			FiNER			FinBERT NER		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
DATE	55.56	83.33	66.67	75.00	100.0	85.71	75.00	100.00	85.71	71.43	83.33	76.92
EVENT	28.57	33.33	30.77	28.57	33.33	30.77	33.33	33.33	33.33	50.00	16.67	25.00
LOC	94.7	92.59	93.63	93.26	93.94	93.60	94.40	92.84	93.62	94.62	90.28	92.39
ORG	66.19	64.65	65.41	85.67	60.86	71.16	92.95	0.6267	74.86	87.69	66.06	75.35
PER	89.44	95.34	92.3	93.27	86.97	90.01	92.05	87.28	89.61	94.13	88.07	91.00
PRO	27.27	30	28.57	53.33	26.67	35.56	50.00	30.00	37.50	32.35	36.67	34.38
MICRO AVG	83.55	84.88	84.21	90.85	81.09	85.70	92.24	81.39	86.48	91.29	81.75	86.26

Table 22: The comparison on the runtimes of the different NER models in seconds.

	Subtitles	Transcriptions	Initialization
Aalto	17.25	12.93	12.25
Lingsoft	17.727	26.176	0
FiNER	808.092	1171.395	0
FinBERT-NER	46.357	73.566	10.291

For Swedish part of the data set, we only tested the Lingsoft NER, as the Aalto, FiNER and the FinBERT NER are only for Finnish. An experimental NER model based on the Swedish BERT is being developed, but at the time of writing, the tagset was experimental and comparison was not really possible. In addition, the Swedish Yle MeMAD NER data set contains Finnish Swedish, and NER systems developed in Sweden for the Swedish names there might not perform very well for names that are from Finland.[12]

[12]Finland is a bilingual country in which most of the towns, public organizations and such have both a Finnish and a Swedish name. For example, Helsinki is Helsingfors in Swedish. Similarly, in bilingual municipalities, street names are often translated. Brand names and such are not, thus a Swedish transcription might also contain Finnish names, which are not necessarily known by a NER model developed in Sweden.

Table 23: The Lingsoft NER results for the Swedish Subtitle set of the Yle test set.

Subtitles	Prec	Rec	F1
DATE	37.93	34.38	36.07
EVENT	00.00	00.00	00.00
LOC	95.09	87.32	91.04
ORG	38.65	62.75	47.84
PER	94.07	92.57	93.32
PRO	00.00	00.00	00.00
Micro avg.	73.20	71.18	72.18

Table 24: The Lingsoft NER results for the Swedish Transcription set of the Yle test set.

Transcriptionss	Prec	Rec	F1
DATE	63.33	38.00	47.50
EVENT	00.00	00.00	00.00
LOC	95.00	85.99	90.27
ORG	53.09	55.07	54.06
PER	95.67	94.14	94.90
PRO	00.00	00.00	00.00
micro avg.	84.74	70.99	77.26

The results for subtitles are shown in Table 23 and for transcriptions in Table 24. The Swedish NER analysis suffers from the lack of EVENT and PRODUCT types, and similarly the recognition results are fairly low for Dates and Organizations as well, and further development for including these types of named entities is required for the Lingsoft NER system.

4.3 NER on ASR

4.3.1 Data

In order to see how well out system performs on ASR output, we tested two datasets, the plenary sessions of the Parliament of Finland sessions or ”Parliament sessions” for short and Yle Pressiklubi data.

The 2017 plenary sessions of the Parliament of Finland contain original video and audio and aligned official transcriptions. It is a public data set downloaded from `https://www.eduskunta.fi` consisting of 5 sessions with 32386 words in total, from which 9370 are unique. The ASR transcripts were provided by the Lingsoft ASR service, which has word error rate of roughly 32.06 %.

The Yle Pressiklubi data set consists of the ASR transcripts of 18 episodes of a Finnish talk show Pressiklubi broadcasted during the period January 1, 2016 and 31 December, 2017. In total, the data set consists of 8 hours, 26 minutes and 42 seconds of programming. The talk show videos are provided to the MeMAD consortium by Yle, and are not available as a public data set. The ASR transcripts were provided by the Lingsoft ASR service. In total, the ASR transcripts of the 18 episodes contain 65422 words.

For these two datasets we do not have gold standard labels, so we used the annotations provided by Lingsoft as true labels. The Lingsoft NER analyses were provided in August 2019, and thus do not include the most recent development in late 2019 reported later in this Deliverable. A repeat study will be carried out in early 2020. The class distribution for the Parliament sessions are shown in Table 25 and for the Yle Pressiklubi in Table 26.

Table 25: The class distribution in Parliament sessions dataset based on the Lingsoft NER analysis

Class	Count
PER	104
LOC	54
TOTAL	158

Table 26: The class distribution in Yle Pressiklubi dataset based on the Lingsoft NER analysis

Class	Count
PER	1350
LOC	601
ORG	327
TOTAL	2278

4.3.2 Results

For the Parliament dataset we have trained Aalto's system on lowercased data and removed the punctuation in order to mimic an ASR setting. Table 27 shows how our system performed when we used Lingsoft annotations as true labels.

Table 27: Results on test set for Parliament Sessions dataset

Entity	Precision	Recall	F1
PERSON	47.87	86.54	61.64
LOCATION	14.06	66.67	23.23
micro avg	28.38	79.75	41.86

We achieved low precision because our system detected a number of entities that were missed by the Lingsoft system.

The number of PERSON entities that were detected by Lingsoft is 104 and the number of LOCATION entities is 54. In order to see how much our system agrees with the Lingsoft system, we evaluated the system only on those entities that were detected by the Lingsoft system. The results of that are presented in Table 28.

Table 28: Results on test set for Parliament Sessions dataset, comparing only entities found by Lingsoft

Entity	Precision	Recall	F1
PERSON	98.90	86.54	92.31
LOCATION	100.00	66.67	80.00
micro avg	99.21	79.75	88.42

Table 29 presents the results for the Yle Pressiklubi dataset where similarly as in the Parliament dataset, we used the Lingsoft annotations as true labels. The number of PERSON entities that were detected by Lingsoft is 1350, number of LOCATION entities is 601 and the number of ORGANIZATION entities is 327. We can notice that the precision for the ORGANIZATION entity is a lot worse compared to the other entities. Next we evaluated only on the entities that were detected by the Lingsoft system to see how much they agree. The results are presented in Table 30.

Table 29: Results on test set for Yle Pressiklubi dataset

Entity	Precision	Recall	F1
PERSON	89.76	84.95	87.29
LOCATION	96.83	89.04	92.77
ORGANIZATION	52.91	44.61	48.48
micro avg	85.06	78.10	81.43

Table 30: Results on test set for Yle Pressiklubi dataset, comparing only entities found by Lingsoft

Entity	Precision	Recall	F1
PERSON	98.37	84.95	91.17
LOCATION	99.24	89.04	93.86
ORGANIZATION	74.29	44.61	55.74
micro avg	95.14	78.10	85.78

The scripts for reproducing the results are available on Github: `https://github.com/aalto-speech/ner-asr`

4.3.3 Conclusion

For the ASR data, we observe that Aalto's system detects more entities than Lingsoft, which results in lower precision when evaluated on datasets that have Lingsoft entities as true labels. On ASR data, Lingsoft analysis sometimes fails to recognize names that are not capitalized by the ASR. In addition, it suffers from the lack of punctuation used as contextual cues. Furthermore we can observe that when we compare the system only on those entities that were detected by Lingsoft, both systems seem to agree, which results in high precision, recall and F1 score.

4.4 Fine-grained Named Entity Recognition

4.4.1 TAC-KBP Entity Discovery and Linking challenge

The goal of TAC-KBP Entity Discovery and Linking (EDL) is to extract mentions of pre-defined entity types, and link (disambiguate and ground) them to the entities in an English knowledge base (KB). In the past several years, the TAC competition has only focused on five major coarse-grained entity types: person (PER), geo-political entity (GPE), location (LOC), organization (ORG) and facility (FAC). Many real world applications in scenarios such as disaster relief and technical support require us to significantly extend the EDL capabilities to a wider variety of entity types (e.g., technical terms, lawsuits, disease, crisis, vehicles, food, biomedical entities). In TAC-KBP2019 the number of types was extended from five to more than 3000 types defined in YAGO[63]. The mention types are organized in a hierarchy (e.g., `Actor` as a subtype of `Artist`, which in turn is a sub-type of `Person`).

There are two stages in EDL– Entity Discovery and Entity Linking.

- **Entity Discovery:** annotators find and annotate mentions for certain kinds of entities that appear in a document.
- **Entity linking:** annotators search through a knowledge base (KB) to determine whether it includes an entry for each entity annotated during Entity Discovery and, if so, link the entity cluster to the KB entry.

4.4.2 Ontonotes dataset

To prepare for the release of the challenge data, we use *Ontonotes*[64], a dataset that is generated through distant supervision. In distant supervision, we make use of an already existing database, such as Freebase or a domain-specific database, to collect examples for the relation we want to extract. We then use these examples to automatically generate our training data. Because these entities are extracted from a KB, it is natural to get more than one type attributed to each entity (Figure 20) and the process is thus susceptible to noisy labels that can be out-of-context or overly-specific for the training sentence. The mention types are hierarchical and are assigned to 89 different tags. Figure 21 shows that the data is very imbalanced where there a disproportionate ratio of observations in each class. this can easily lead any model to be biased by the class which has a lot of examples.

4.4.3 Approach

To tackle the task, three methods can be considered:

- **"Flat" Classification Approach:** is the simplest way to deal with hierarchical classification problems. It consists of ignoring the class hierarchy, typically predicting only classes at the leaf nodes. However, this very simple approach has the serious disadvantage of having to build a classifier to discriminate among a very large number of classes, without exploring information about parent-child class relationships present in the class hierarchy.
- **Local Classifiers Approach:** The local classifiers approach consists of training one multi class classifier for each node of the mention types hierarchy. the main issue for this solution is the big number of models to train and the error committed in the parent node will be propagate to the ancestor nodes.
- **Classification with Hierarchy-Aware Loss:** The solution consist of adopting the idea of hierarchical loss function[65] to adjust the penalties for Fine-grained Entity Type Classification depending on how far they are in the hierarchy. For example, the penalty for predicting `Person` instead of `Person/Athlete` should less than the penalty for predicting `Organization`.

Our solution relies a two stage model: we use BERT to predict the first level of the hierarchy then, we extract the BERT embedding and concatenate it with the probability output of the BERT classifier (the output used to predict the first level of the hierarchy) in order to train the classifier which will predict the deeper level of classification.

Because of the automatic annotation process, we may have multiple labels for the same annotated entity (DBPedia does not limit the number of types an entity can have). We preprocess the data in the following fashion:

1. Extract the named entity using the "start" and "end" features.
2. Keep the most *probable* mention type for each named entity recognition. We define this probability by counting by choosing the deepest type associated to the most frequent category. We can illustrate this process with an example: for one named entity we can get the following tags: `[/Person, /Person/Artist, /Person/Artist/Actor, /Other/Art]`, the mention we keep for this example is `/person/artist/actor`, which is the longest in the hierarchy and is associated to `/person` (Which is more frequent than `/Other`), so it's more probable for this named entity to be a `Person/Artist/Actor`.
3. Convert the Ontonotes data to the CoNLL format.

We also split the data into 3 subsets to train the two-stages model:

1. Training dataset I: used to fine-tune the BERT model to predict the first level of the tag hierarchy.

2. Training dataset II: Train the model of the next stage (because we use the prediction of the model of the first stage as an input to the model of the second stage). Doing this reduces the risk of overfitting, especially for classes with very small support.

3. Test dataset: used to test the whole stage, and this data should not be used to train any previous model.

4.4.4 The model

The performance of the model relies on having good representations for the data points we try to classify. For a long time, the NLP community relied on building a model on top of pre-trained Word Embeddings such as Word2vec[66] and GloVe[67]. But since the popularity of BERT[47], we chose it as the base component of our model (figure 23). BERT stands for "Bidirectional Encoder Representations from Transformers", and is currently the state-of-the-art language model for most NLP tasks. BERT is designed to produce deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, BERT can be fine-tuned by just adding an output layer and it is able to outperform all the existing models. Figure 24 shows a simplified diagram of BERT's architecture.

We use BERT here to predict the first level of the tags hierarchy. If BERT predict that the tag is "O" (for `Other`, or "not a named entity"), we don't need to pass the corresponding token to the classifier head to predict the next level. In contrast, if BERT predicts that the corresponding token is an entity, we extract BERT embedding and feed it to our classifier to predict the full hierarchy mention type.

The model is comprising of two components:

1. **Pre-trained BERT**: We fine-tune the pre-trained BERT model (`bert-base-cased`) using the `training dataset I` after converting it to the CoNLL format, first to learn a representation for each token and the context (surrounding words), and second to learn a classifier for the first level in the hierarchy of tags as well as the *Not an entity* class.

2. **A classifier head**: after using BERT to generate an embedding for each token, its context and a class distribution, we concatenate them to train the classifier. We use the `training dataset II` to train a classifier to predict the sub-tag for each token.

Contextual embedding:

Embeddings are dense vector representations of words in lower dimensional space. Using such representations allow the model to leverage the semantics of a word in a numerical form by performing mathematical operations on it. In Figure 25, we can see a breakdown of the input to the BERT model.

BERT utilizes WordPiece for breaking down sentences into tokens (*tokenization*), which splits tokens like `\playing"` to `\play"` and `\##ing"`. This mainly solves the Out-Of-Vocabulary (OOV) problem. We extract the contextual BERT embedding from the last layer.

Multi-class Classifier head

This stage aims to predict the full hierarchy mention type, using BERT embedding and the probabilities assigned to each class (here we have 8 different class `[B-person, I-person, B-organization, I-organization, B-other, I-other, B-location, I-location]`). The probabilities aim just to give the classifier information about the first level, so i can get the scenario where BERT predict for a token that is a location (a wrong classification done at the first level) but the classifier can correct this mistake predict that the same token is a ”/person-/artist”.

The main challenge here that the data is very imbalanced, as there are some mention types with just few examples (e.g. ”person/legal”, ”person/military”, ”organization/transit”, ”other/award”).

4.4.5 Results

In this section, we present the obtained results from our model using different classifier heads for the different data partitions. The dataset contains a total of 85 844 sentences.

Coarse-grained NER Result

Table 31 shows the the results obtained at the first layer of the classifier. We fine-tune the pre-trained `"bert-base-cased"` BERT model for 10 epochs.

We can see from the Table 31 that we don’t need a lot of data to get a good result. For different train/test partition, the result is almost the same. However, to train the final classifier classifier, we need a lot of Data.

Train/test splits : number of sentences	metric	person	location	other	organization
68 675/17 169	precision	0.7634	0.8356	0.7881	0.7335
	recall	0.7956	0.8377	0.7908	0.7437
	f1-score	0.7792	0.8367	0.7895	0.7386
51 507/34 337	precision	0.7751	0.8349	0.7985	0.7554
	recall	0.7927	0.8440	0.7951	0.7537
	f1-score	0.7838	0.8394	0.7968	0.7546
17 169/68 675	precision	0.7634	0.8190	0.7834	0.7263
	recall	0.7723	0.8375	0.7809	0.7351
	f1-score	0.7678	0.8281	0.7822	0.7306

Table 31: BERT-base Result for different data partition

We fine-tune the larger `"bert-large-cased"` pre-trained model for the same amount of epochs, and the table 32 present the obtained result and it is almost the same as BERT-base model.

We see that the choice of the pretrained model (`base` and `large`) doesn’t affect the results significantly.

Train/test partition : number of sentences	metric	person	location	other	organization
68 675/17 169	precision	0.7808	0.8232	0.7982	0.7396
	recall	0.7683	0.8095	0.7620	0.7067
	f1-score	0.7745	0.8163	0.7797	0.7228

Table 32: BERT-Large Result

Fine-grained NER Result

In this section, we experiment with the different classification layers built on top of the BERT base. We got the best performances using SVM, a logestic regression and XGBClassifier (One versus Rest), as the results can be seen in Table 33. We also experiment with different splits for the training data reserved for fine-tuning the BERT model, and the portion used to train the classifier, as we observe that fine-tuning the BERT model can easily lead to overfitting.

Train/test partition : number of tokens	metric	SVM	Logestic	XGBClassifier
33 074 / 8 269	micro-f1	**0.54**	0.45	0.51
	macro-f1	**0.26**	0.23	0.24
66 242 / 16 562	micro-f1	**0.58**	0.50	0.58
	macro-f1	**0.33**	0.23	0.32
99 241 / 24 812	micro-f1	**0.57**	0.46	0.54
	macro-f1	**0.37**	0.27	0.29

Table 33: Classifiers results for different data partitions

The figure 26 presents the breakdown of results per class. We can generally note that the model for most classes boasts high precision but low recall scores, thus damaging the final F1-score. SVM also performs the best among these linear classifiers.

Train/test partition	metric	CNN	DNN
33 074 / 8 269	micro-f1	0.49	0.54
	macro-f1	0.24	0.32
66 242 / 16 562	micro-f1	0.52	**0.57**
	macro-f1	0.28	**0.36**
99 241 / 24 812	micro-f1	0.49	0.54
	macro-f1	0.28	0.33

Table 34: Deep learning classifiers Results for different data partition

Finally, we experiment with adding deeper classifiers on top of BERT contextual embeddings, namely a vanilla 2-layers Dense Neural Network and a Convolutional Neural Network (CNN). The architecture of both models can be found in figures 27 and 28, respectively. Table 34 shows the results obtained by these two architectures.

The best model achieves a 0.57 micro/0.46 macro F1-score on the data, which shows there is still room for improvement. The hierarchy of the classes can be exploited further, either by creating better local classifiers or through exploiting a hierarchical loss during training.

4.5 EURECOM NER on French ASR

To better understand the distribution of Named Entities on our French corpus, we administer an exhaustive extraction on the automatically generated subtitles for the INA Professional Archive content, amounting to a total of 490 hours of audio content. The entity extraction is performed using SpaCy[13], an LGPL-licensed model trained on the WikiNER corpus[14]. This model classifiers NEs into 4 categories: `PER` (person), `ORG` (organization), `LOC` (location), and `MISC` (miscellaneous).

[13] `https://spacy.io/models/fr`
[14] `https://figshare.com/articles/Learning_multilingual_named_entity_recognition_from_Wikipedia/5462500`

4.5.1 Breakdown by Publication Channel and Entity Type

In Table 35, we break down the distribution of Named Entities by channel and entity type. We notice that the ratios of entity types are consistent across the corpus: around a third of recognized entities are People, followed by a comparable number of Locations, then Miscellaneous and Organizations at about 20% and 10% respectively. Although the quantity of available content from all channels is comparable, we see that there are much more entity mentions on Radio Channels than on TV channels, probably due to the nature of the medium (while we can rely on images to identify people and concepts on TV, they have to be explicitly and maybe repeatedly mentioned on radio for the listeners).

The distribution of the recognized entities is also consistent with the training set used for training the model: 31.5%, 37.7%, 11.0%, 19.8% for `PER`, `ORG`, `LOC`, and `MISC`, respectively.

Channel	Type	# Programs	# Hours	# Entities	PER	LOC	ORG	MISC
"France 2"	TV	235	130.49	39783	37.43%	31.4%	9.14%	22.03%
"France Inter"	Radio	550	182.235	56560	37.48%	34.73%	9.1%	18.68%
"France Culture"	Radio	302	178.91	73292	37.57%	31.81%	12.97%	17.65%
Total		1088	491.64	169635	37.51%	32.69%	10.78%	19.02%

Table 35: Breakdown of NER results by Publication Channel and Entity Type

4.5.2 Breakdown by Genre

In table 36, we break down the distribution of Named Entity by genre. There is a noticeable variety in term of Named Entity density (number of mentions per unit of time). Genres that relate to actuality and news tend to have a significantly higher number of entity mentions (peaking at 625 mentions per hours, or 10.4 mentions per minutes), while general audience and entertainment programs such as Documentaries and Game Shows do not feature as much references to named entities (as low as 236 mentions per hour, or 3.9 mentions per minutes). The difference is sufficiently significant to prove a tight correlation between the genre of the studied program and the amount of real world-grounded knowledge that can be extracted from it.

Again, we notice that radio content is has a significantly higher named entity mentions per time ratio. While TV content clocks at 374 mentions per hour (6.25 mentions per minute), radio/audio content boasts a density of 520 mentions per hour (or 8.66), almost 1.4 times as much as audiovisual content, across genres.

Genre (fr)	Genre (en)	# Programs	%	# of hours	# NE	Density
"Entretien"	Interview	288	20.75%	169.78	57931	341.21
"Magazine"	Magazine	219	15.78%	169.09	53044	313.70
"Reportage"	Report	80	5.76%	37.94	10306	271.64
"Jeu"	Game Show	43	3.10%	23.72	6866	289.46
"Débat"	Debate	33	2.38%	23.06	8463	367.00
"Journal parlé"	Radio News	88	6.34%	19.88	12435	625.50
"Documentaire"	Documentary	21	1.51%	16.84	3986	236.70
"Chronique"	Chronicle	253	18.23%	15.46	6744	436.22
"Journal télévisé"	Televised News	50	3.60%	14.74	6319	425.81
"Spectacle radio"	Radio Show	19	1.37%	9.98	3674	368.14
"Lecture"	Lecture	21	1.51%	6.51	2189	336.25
"Revue de presse"	Press Review	44	3.17%	2.60	1565	599.62
	Other genres	29	2.06%	18.07	63455	315.08
	Total TV	235	21.61%	130.49	48852	374.37
	Total Radio	852	78.39%	361.15	188125	520.90

Table 36: Breakdown of NER results by Genre

4.5.3 A closer look

We analyze up close the results we get from running NER on our corpus. We extract the named entities from two sources: the textual description in the metadata (summary) and the automatically generated subtitles. This was done on a 5 minutes speech presented by the French president François Holland after the announcement of the European Elections results on May the 26th 2014. The metadata on this program can be found here: `http://data.memad.eu/fr2/orphan/1e3a3191c667f08782f571ff1e53145b5c7432a4`. Figure 29 shows a screenshot of the program on the Flow platform.

The particularity of this program is that the transcription of the speech is entirely provided in the "Summary" metadata field, which allows us to compare the results on the transcription against the results on the ASR, which lack capitalization, punctuation and contain several transcription errors.

Source	Type	Extracted Entities
Title	PER	"Monsieur François Hollande"
Summary	PER	"Manuel VALLS", "Président de la République"
	LOC	"Elysée", "Etats", "Euro", "Europe", "Européen", "France", "Français", "République", "l'Europe", "la France"
	ORG	"Commission Européenne", "Conseil européen", "Patrie des droits de l'homme", "Union Européenne"
	MISC	'Dimanche", "Français", "Mes chers compatriotes", "Monsieur François HOLLANDE", "République", "Source : Vie"
Subtitles	PER	"manuel valls'
	LOC	"l'europe", "france", "europe", "français", "la france"
	ORG	"l'extrême droite', "gérard", "union européenne patrie des droits de l'homme"

Table 37: Extracted entities from *"Déclaration du Président de la République, Monsieur François Hollande"*

The model is able to detect both proper names e.g. "Manuel Valls", "Monsieur François Macron" and position titles "Président de la République" (tr. *President of the Republic*) as PERSON. It also correctly classifies place names as well as demonyms such as "European" and "French" as LOCATION, even correctly labeling "Euro" as such in the sentence "la zone Euro" (tr. *the Euro Zone*), without confounding it with the currency. It also does a good job extracting organizations such as "Union Européen" and "Conseil Européen" but only when properly capitalized. In "MISC", it generates a lot of false entities because of the capitalization of some words. It also fails to recognize events such as "élections européennes" (tr. *European Elections*), "la crise de la zone Euro" (tr. *Euro Zone crisis*), or key concepts such as "droits de l'homme",

”mondialisation”, ”transition énergétique”, ”justice sociale”, ”éducation” (tr. *”Human rights”, ”globalization”, ”energy transition”, ”social justice” and ”education”*, respectively), which can be very important in the tasks of indexing and retrieval of relevant content from the corpus. This points out the need to develop a system that is capable of detecting more fine-grained categories such as Events and Concepts, which is not available in off-the-shelf NER classifiers to date, especially for non-English langauges.

4.5.4 Wikifier output

Since our interest may lay beyond named entities but any entity or concept that can be of interest to archiving or retrieving the content, we use Wikifier[15], an online multilignal service that can tag spans of text corresponding to concepts or entities with Wikipedia articles and link them to it (joint entity extraction and linking). Figure 30 shows the output generated from the ASR transcription of the speech.

As we can see from the picture, Wikifier succeeds in extracting a much more rich set of entities from the text, not being confined to the 4 original NER tags. Concepts and Events such as the aforementioned references to the *Euroozone crisis* and *European Elections* are not only highlighted but also properly linked to their corresponding Wikipedia page (with decreasing order of confidence, if there are many). This approach seems to be more promising for the purposes of tagging and classifying the available content, especially enabling the possibility of linking it further to other similar resources, both inside the knowledge graph and to external resources.

It’s worth noting that Wikifier can be more or less strict in extracting entities and links by adjusting its confidence threshold, the trade-off being that with increased recall (more text spans being linked to their respective Wikipedia page) precision usually takes a dip, injecting more noise in the process.

4.6 Aligning ASR with manual subtitles

4.6.1 The alignement process

In the context of the project, both Automatic Speech Recognition results as well as manually-written subtitles are provided for some programs. But aligning the two to study the artifacts of ASR (on the process of named entity extraction for example) proved to be quite challenging, mainly because of (1) the timing differences between the automatic and human process (resulting in one being in advance or belated compared to the other), (2) the errors in the ASR and the shortcuts taken by human annotators and (3) the length of each subtitle, as human annotators tend to break down the sentences in a more deliberate and conceptual way, whereas the automatic methods generally use the sentence length as a criterion to break down the transcribed text. In this section we describe the process of aligning text from both sources.

The process works as follows (figure 31):

1. Pick one line from the subtitles which starts at time `t`
2. Retrieve all lines from the ASR data within a range of `[t-2min,t+2min]` (to account both for timing inaccuracies as well as line lengths)
3. Calculate a string similarity score between the original content and every retrieved line
4. Keep the pair with greatest score

[15] `http`

To compute the string similarity score, we use the FuzzyWuzzy[16] python package to compute the Levenshtein Distance between the two strings, i.e. the minimum number of edits (insertions, deletions or substitutions) that need to be done to change a one sequence into the other.

The figure 32 represents the matching score values distribution (time interval search=[t-2min,t+2min]). We should set a threshold to accept or not the corresponding pairs (the alignment), which we fix empirically at 51. We repeat the same process again with a larger time interval.

As a result, we could align 9190/24327 (37.7%) for the subtitle data and 9190/51387 (17.8%) for the ASR data.

The reason for this low coverage is the discrepancy in the timing chosen to break the sentences in the two streams of subtitles. A fix to this would be to try and match the entire textual content (every line) from both sources and then choose the timing provided by either sources.

4.6.2 NER on the aligned corpora

We present the result obtained by using SpaCy for both the aligned ASR and video subtitles data. The figure present some examples of the aligned data and the extracted named entity.

data	LOC	PER	ORG	MISC
Subtitles	5322	4375	1056	1420
ASR	5409	4836	1306	1375

Table 38: comparison between the number of extracted named entity from the ASR data and the subtitle data

Upon investigating the results we got by running a NER system on both data streams, we notice several remarks:

1. The Subtitles explicitly mention the speaker sometimes (to disambiguate in case of multiple people talking) e.g. ”Pierre Croce : Non. C’est plus classique.”, with ”Pierre Croce” being the speaker name (PER).

2. The punctuation marks added sometimes by human annotators help the model better identify some named entities, e.g. ”Slaviansk, encerclée par l’armée ukrainienne..” (tr. *”Slovyansk, surrounded by the ukranian army..”*), the model successfully recognizes ”Slaviansk” as a LOC on the subtitles data, but fails to do so on ASR.

3. When the model encounters capitalized second-person plural verbs, e.g. *Allez, Continuez, Laissez* (tr. *”(You) Go, continue, leave”*) it tends to classify them as PER, given the frequency of last names ending with *-ez* and are usually capitalized.

4. The annotators sometimes use some shortcuts to make the lines shorter by dropping a first name or using an abbreviation instead of the full words, e.g. *F.N. and U.E.* instead of Front National and Union Européenne, making it harder for the classifier to correctly predict the corresponding tag.

[16] `https://github.com/seatgeek/fuzzywuzzy`

5 Conclusion

In this deliverable, we have first presented the evolution of the MeMAD knowledge graph. In particular, we have integrated some first results of the multimodal analysis results performed in WP2 such as automatic speech recognition (ASR) or visual face detection and recognition results. We proposed a generic method enabling to generate Web APIs on top of any RDF-based knowledge graph. This method has been used to generate the MeMAD knowledge graph API which has been successfully used to integrate all legacy metadata from the content provider in the Flow platform.

Next, we have researched and developed methods to predict the interestingness or memorability of a given sequence, named moment, from multimedia content. This task being, by nature, highly subjective, we have relied on standard corpora. First, we participate in a competitive MediaEval task where the goal is predict the short term and the long term memorability of some short videos. Our approach is multimodal, exploiting on one side visual and audio features and on the other side, textual features extracted from short descriptions but also automatically generated deep captions in a sort of data augmentation manner. Our approach was comparatively successful since we obtained the best score on this task with respect to all other participants. Next, we have decided to tweak and apply this method on longer videos, which represent the MeMAD use cases. We rely here on the Cognimuse dataset which provides labelled data about interesting sequences that need to be predicted. At the time of writing, the results are being computed. We plan to use this method on the MeMAD dataset and to run a qualitative evaluation during the third and final year of the project.

Once detected, moments benefit from being enriched with additional information which can also be used for searching particular moments. This typically relies on annotating those moments with entities defined in general purpose knowledge graph such as Wikidata. We have further developed a number of NER approaches, either relying on rules (Lingsoft) or on pure deep learning algorithms (Aalto and EURECOM), using modern transformers-based contextual word embeddings (BERT). The scientific challenges we have tackled are numerous:

- Multilinguality: we have evaluated our approaches on French and Finnish, beyond English which is traditionally used in benchmarks;
- Types granularity: we have evaluated our approaches on a very large number of entity types including concepts as this is more realistic to MeMAD scenarios;
- Noisy input texts: we have evaluated our approaches on ASR rather than clean text, considering that ASR are by nature imperfect, may contain grammatical errors and generally lack good casing and punctuations which are critical for natural language processing.

Those results will be further consolidated during the third year of the project. In particular, we will also investigate how we could extract named entities directly from the audio without using the textual modality in an end-to-end approach.

6 References

[1] Danny Francis, Benoit Huet, and Bernard Merialdo. EURECOM participation in TrecVid VTT 2018. In *22nd International Workshop on Video Retrieval Evaluation (TRECVID)*, Gaithersburg, USA, 2018.

[2] Pasquale Lisena, Albert Mero no Peñuela, Tobias Kuhn, and Raphaël Troncy. Easy Web API Development with SPARQL Transformer. In *International Semantic Web Conference (ISWC)*, pages 454–470, 2019.

[3] Ismaïl Harrando, Benoit Huet, Raphaël Troncy, Jean Carrive, Steffen Lalande, Michael Stornbom, Tiina Lindh-Knuutila, Lauri Saarikoski, and Kim Viljanen. D3.1: TV programme annotation model. Technical report, December 2018.

[4] Karel Braeckman, Simon Debacq, Harri Kiiskinen, Nico Oorts, Lauri Saarikoski, Raphaël Troncy, Wim Van Lancker, Dieter Van Rijsselbergen, Maarten Verwaest, and Kim Viljanen. D6.4: Specification of the data interchange format, intermediate version. Technical report, June 2019.

[5] Pasquale Lisena and Raphaël Troncy. Transforming the JSON Output of SPARQL Queries for Linked Data Clients. In *International World Wide Web Conference (TheWebConf), Developers Track*, pages 775–780, 2018.

[6] 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.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[8] 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.

[9] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR*, abs/1506.06724, 2015.

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[11] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, 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.

[12] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015.

[13] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[14] Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline, 2019.

[15] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

[16] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *CoRR*, abs/1904.01766, 2019.

[17] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations, 2019.

[18] Mihai Gabriel Constantin, Miriam Redi, Gloria Zen, and Bogdan Ionescu. Computational understanding of visual interestingness beyond semantics: literature survey and analysis of covariates. *ACM Computing Surveys (CSUR)*, 52(2):1–37, 2019.

[19] Mohammad Soleymani. The quest for visual interest. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 919–922, 2015.

[20] Zoya Bylinskii, Michelle A Borkin, Nam Wook Kim, Hanspeter Pfister, and Aude Oliva. Eye fixation metrics for large scale evaluation and comparison of information visualizations. In *Workshop on Eye Tracking and Visualization*, pages 235–255. Springer, 2015.

[21] Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What makes a photograph memorable? *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1469–1482, 2013.

[22] Mengjuan Fei, Wei Jiang, and Weijie Mao. Creating memorable video summaries that satisfy the user's intention for taking the videos. *Neurocomputing*, 275:1911–1920, 2018.

[23] Claire-Hélène Demarty, Mats Sjöberg, Bogdan Ionescu, Thanh-Toan Do, Michael Gygli, and Ngoc Duong. Mediaeval 2017 predicting media interestingness task. In *MediaEval workshop*, 2017.

[24] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. *CoRR*, abs/1801.07424, 2018.

[25] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder-decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[26] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Asian Conference on Computer Vision*, pages 39–54. Springer, 2018.

[27] Kaiyang Zhou and Yu Qiao. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. *CoRR*, abs/1801.00054, 2018.

[28] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015.

[29] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[30] Mihai Gabriel Constantin, Bogdan Ionescu, Claire-Helene Demarty, Ngoc Q. K. Duong, Xavier Alameda-Pineda, and Mats Sjoberg. The predicting media memorability task at mediaeval 2019. *Proc. MediaEval workshop*, 2019.

[31] Rohit Gupta and Kush Motwani. Linear models for video memorability prediction using visual and semantic features. In *MediaEval*, 2018.

[32] Duy-Tue Tran-Van, Le-Vu Tran, and Minh-Triet Tran. Predicting media memorability using deep features and recurrent network. In *MediaEval*, 2018.

[33] 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.

[34] 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, 2016.

[35] Jianxiong Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010.

[36] 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.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[38] Yuncheng Li, Yale Song, Liangliang Cao, Joel R. Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A new dataset and benchmark on animated GIF description. *CoRR*, abs/1604.02748, 2016.

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[40] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016.

[41] Huet-B. Francis, D. L-stap : Learned spatio-temporal adaptive pooling for video captioning. In *First International Workshop on AI for Smart TV Content Production (AI4TV)*, 2019.

[42] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[43] Yoon Kim. Convolutional neural networks for sentence classification. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[44] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[45] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark`, 2018. Accessed: [08.26.2020].

[46] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[47] 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.

[48] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–883, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[49] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016.

[50] Athanasia Zlatintsi, Petros Koutras, Georgios Evangelopoulos, Nikolaos Malandrakis, Niki Efthymiou, Katerina Pastra, Alexandros Potamianos, and Petros Maragos. Cognimuse: a multimodal video database annotated with saliency, events, semantics and emotion with application to summarization. *EURASIP Journal on Image and Video Processing*, 2017(1):54, 2017.

[51] Petros Koutras, Athanasia Zlatinsi, and Petros Maragos. Exploring cnn-based architectures for multimodal salient event detection in videos. In *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. IEEE, 2018.

[52] Petros Koutras and Petros Maragos. Susinet: See, understand and summarize it. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[53] Olfa Ben-Ahmed and Benoit Huet. Deep multimodal features for movie genre and interestingness prediction. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2018.

[54] Tulika Saha, Aditya Patra, Sriparna Saha, and Pushpak Bhattacharyya. Towards emotion-aided multi-modal dialogue act classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4361–4372, 2020.

[55] Dejan Porjazovski, Juho Leinonen, and Mikko Kurimo. Named entity recognition for spoken finnish. In *2nd International Workshop on AI for Smart TV Content Production: Affiliation; Access and Delivery*, New York, NY, USA, 2020. ACM.

[56] 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.

[57] Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, 2014.

[58] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[59] K. Koskenniemi. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. PhD thesis, University of Helsinki, Dept of General Linguistics, Finland., 1983.

[60] F. Karlsson. *Constraint Grammar: A Language-Independent Framework for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin / New York, 1995.

[61] Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, pages 1–26, 2019.

[62] Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: Bert for finnish, 2019.

[63] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[64] Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, et al. Ontonotes release 2.0.

[65] Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. *arXiv preprint arXiv:1803.03378*, 2018.

[66] 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.

[67] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

A Dissemination activities

- **Talk** 12/06/2019: MDN 2019: *EBU Metadata Developer Network Workshop*, Geneva, Switzerland. Ismail Harrando
- **Workshop organization** 21/10/2019: AI4TV 2019: *1st International Workshop on AI for Smart TV Content Production, Access and Delivery*, a workshop at ACM International Conference on Multimedia, Nice, France. Raphaël Troncy and Jorma Laaksonen chaired the workshop.
- **Keynote** 22/10/2019: ACMMM19: *ACM International Conference on Multimedia*, Nice, France. Jean Carrive presented the keynote *Using Artificial Intelligence to Preserve Audiovisual Archives: New Horizons, More Questions*.
- **Workshop presentation** 27/10/2019: MediaEval 2019: *MediaEval Benchmarking Initiative for Multimedia Evaluation*, Sophia Antipolis, France. Alison Reboud and Ismail Harrando presented *Combining textual and visual modeling for predicting media memorability*.
- **Conference presentation** 29/10/2019: ISWC 2019: *The 18th International Semantic Web Conference*, Auckland, New Zealand. Pasquale Lisena presented *Easy Web API Development with SPARQL Transformer*.
- **Workshop presentation** 03/12/2019: SemWebPro 2019: *Journée de présentations et de rencontres dédiées au web sémantique dans le monde professionnel*, Paris, France. Pasquale Lisena presented *Easy Web API Development with SPARQL Transformer*.
- **Workshop presentation** 13/12/2019: Corpus Workshop at BnF: Jean Carrive presented *New Analysis Methods for Audiovisual Media: ANTRACT and MeMAD projects*, Collect, Preserve, Explore Massive Audiovisual Corpora Workshop, National Library of France (BnF)

B Appendices

B.1 EURECOM MDN 2019 Talk

In the context of the European research project MeMAD (Methods for Managing Audiovisual Data), we face the challenge of modeling semantically audiovisual legacy metadata and results of automatic analysis from multiple partners and in an interoperable manner.

In this talk, we present an implementation of the EBU-CCDM/EBU Core data model for representing production and broadcasting information of TV and Radio programs provided by two partners, namely INA in France and Yle in Finland. The so-called resulting MeMAD knowledge graph provides metadata for more than 60K hours of audiovisual content, spanning multiple channels, audiovisual genres, themes and languages.

We give a quantitative overview of the data in terms of size and scope, its original format as well as the working RDF model into which all data has been converted. We present several controlled vocabularies and alignments attempts to enrich the data. We describe how results of automatic analysis algorithms (e.g. face recognition, speaker diarization, named entity recognition and disambiguation, automatic speech recognition, etc.) can also be materialized and queried in this knowledge graph. Finally, we show how this knowledge graph can be accessed, using either SPARQL as an API or via a dedicated REST-based API automatically generated.

B.2 EURECOM and AALTO's MediaEval 2019 workshop paper [1]

This paper describes the models that the EURECOM and AALTO teams submitted to MediaEval 2019 Media Memorability Track and summarises their results.

Combining Textual and Visual Modeling for Predicting Media Memorability

Alison Reboud[*], Ismail Harrando[*], Jorma Laaksonen[+],
Danny Francis [*], Raphaël Troncy[*], Héctor Laria Mantecón[+]

[*]EURECOM, Sophia Antipolis, France
[+]Aalto University, Espoo, Finland
{alison.reboud,ismail.harrando,danny.francis,raphael.troncy}@eurecom.fr
{jorma.laaksonen,hector.lariamantecon}@aalto.fi

ABSTRACT

This paper describes a multimodal approach proposed by the MeMAD team for the MediaEval 2019 "Predicting Media memorability" task. Our best approach is a weighted average method combining predictions made separately from visual and textual representations of videos. In particular, we augmented the provided textual descriptions with automatically generated deep captions. For long term memorability, we obtained better scores using the short term predictions rather than the long term ones. Our best model achieves Spearman scores of 0.522 and 0.277 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 long term memorability score for a given set of annotated videos. The full description for this task is provided in [2]. Last year's best approaches for both the long term[5] and short term tasks [14] indicated that high level representations extracted from deep convolutional models performed the best in terms of visual features. Furthermore, the best long term model [5] was a weighted average method including Bag-of-Words features extracted from the provided captions. Following this approach, we created multimodal weighted average models with visual deep features and textual features extracted from both the provided video titles, as well as from automatically generated deep captions.

2 APPROACH

2.1 Visual Approaches

VisualScore. Our visual-only memorability prediction scores are based on using a feed-forward neural network with visual features in the input, one hidden layer of 430 units and one unit in the output layer. The best performance was obtained with 6938-dimensional features consisting of the concatenation of I3D [1] video features, ResNet-152 and ResNet-101 [6] image features and two versions of SUN-397 [15] concept features. The image and concept features were extracted from the middle frames of the videos. The hidden layer uses ReLU activations and dropout during the training phase, while the output unit is sigmoidal. We trained separate models for the short and long term predictions with the Adam optimizer. The number of training epochs was selected with 10-fold cross-validation with 6000 training and 2000 testing samples.

CaptionsA. Our first captioning model uses the DeepCaption software[1] and is quite similar to the best-performing model of the PicSOM Group of Aalto University's submissions in TRECVID 2018 VTT task [13]. The model was trained with COCO [10] and TGIF [9] datasets using the concatenation of ResNet-152 and ResNet-101 [6] features as the image encoding. The embed size of the LSTM network [7] was 256 and its hidden state size 512. The training used cross-entropy loss.

CaptionsB. Our second model has been trained on the TGIF [9] and MSR-VTT [16] datasets. First, 30 frames have been extracted for each video of these datasets. Then, these frames have been processed by a ResNet-152 [6] that had been pretrained on ImageNet-1000: we keep local features after the last convolutional layer of the ResNet-152 to obtain features maps of dimensions 7x7x2048. At that point, videos have been converted into 30x7x7x2048-dimensional tensors. A model based on the L-STAP method [4] has been trained on MSR-VTT and TGIF: all videos from TGIF, and training and testing videos from MSR-VTT have been used for training, and validation has been performed throughout training with the usual validation set of MSR-VTT, containing 497 videos. Cross-entropy has been used as the training loss function. The L-STAP method has been used to pool frame-level local embeddings together to obtain 7x7x1024-dimensional tensors: each video is eventually represented by 7x7 local embeddings of dimension 1024. These have been used to generate captions as in [4].

VisualEmbeddings. The local embeddings used for CaptionsB have also been used to derive global video embeddings, by averaging the mentioned 7x7 local feature embeddings. These global video embeddings have then been fed to a model of two hidden layers, the first one and the second one having respectively 100 and 50 units, and ReLU activation function. The number of training epochs is 200 with an early stopping monitor.

MediaEval'19, 27-29 October 2019, Sophia Antipolis, France

[1]https://github.com/aalto-cbir/DeepCaption

2.2 Textual Approaches

Through initial experiments and from last year's results on this task, the descriptive titles provided with each video prove to be an important modality for predicting the memorability scores. In order to build on this observation, we generate captions for each video using the two visual models described above (**CaptionsA** and **CaptionsB**). While the generated captions are not always accurate, they seem to noticeably help the model disambiguate some titles and use some of the vocabulary already seen on the training set (e.g. the title contains words such as *couple"* or *"cat"* while the generated caption would say *"a man and a woman"* or *"an animal"*, respectively, which are more common words in the training set and thus help the model generalize better on inference time). The models described in this section use a concatenation of the original provided title and the generated captions as their input.

Multiple techniques for generating a numerical score from this input sequence were considered (in ascending order of their performance on cross-validation).

Recurrent Neural Network. We use an LSTM [7] to go through the GloVe embeddings [12] of the input and predict the scores at the last token. This model performed consistently the worst, probably due to the length of the input sequence at times, and the empirical observation that word order doesn't seem to matter for this task.

Convolutional Neural Network. We use the same model as [8] except for a regression head instead of a classifier trained on top of the CNN, and GloVe embeddings as input. This model leaks less information thanks to max-pooling, and performs much better than its recurrent counterpart.

Self-attention. Similar to the previous methods, we feed our input text to a self-attentive bi-LSTM [11] to generate a sentence embedding that we use to predict the memorability scores. This model performs on par with the CNN method.

BERT. We used a pre-trained BERT model [3] to generate a sentence embedding for the input by max-pooling the last hidden states and reducing their dimension through PCA (from 768 to 250). This model performs better than the previous ones but it is more computationally demanding.

Bag of Words. We vectorize the input string by counting the number of instances of each token (and frequent n-grams) after removing the stop words and the least frequent tokens. The score is predicted by training a linear model on the counts vector. This simple model performs the best on our cross-validation, which can be justified by the lack of linguistic or grammatical structure in the titles and generated captions that would justify the use of a more sophisticated model.

For all the models considered, the addition of the generated captions improves the prediction score on the validation set considerably. It also should be noted that the use of short-term scores for long-term evaluation yields substantially better results throughout all of our experiments.

3 RESULTS AND ANALYSIS

During the evaluation process, we created four test folds of 2000 videos and therefore four models trained on 6000 videos. For the VisualScore approach, we decided to use predictions from a model trained on the entire set of 8000 videos (VisualScore8k), as well as the mean predictions from the combinations of the four models trained on 6000 videos (VisualScore6k). For the Long Term task, all models except from the WA3lt exclusively use short-term scores.

Table 1: Results on test set for short term memorability

Method	Spearman	Pearson	MSE
Textual	0.441	0.464	0.01
VisualScore	0.495	0.543	0
WA1	0.512	0.552	0
WA2	**0.522**	**0.559**	0
WA3	0.520	0.557	0

Table 2: Results on test set for long term memorability

Method	Spearman	Pearson	MSE
Textual	0.239	0.25	0.03
VisualScore	0.268	0.289	0.03
WA2	**0.277**	**0.296**	0.03
WA3	0.275	0.295	0.03
WA3lt	0.260	0.285	0.02

- WA1 = 0.5Textual+0.5VisualScore
- WA2 = 0.25Textual+0.25VisualEmb+0.5VisualScore8k
- WA3 = 0.25Textual+0.25VisualEmb+0.5VisualScore6k
- WA3lt = WA3 with long-term scores

We observe that the weighted average method which was trained on the whole training set and included our two visual approaches and our textual approach works the best for short term predictions. For long term prediction, one of the key observations to make is that WA3lt got the second worst results. This is consistent with our early observation that short-term scores for long-term evaluation yields substantially better results.

4 DISCUSSION AND OUTLOOK

This paper describes a multimodal weighted average method outperforming the best results of the Predicting Media Memorability Task 2018. One of the key contribution of this paper is to have demonstrated that using deep captions helped improving the predictions. We also conclude that, quite surprisingly, a simple n-gram frequency count was more efficient at modelling memorability than more sophisticated textual models. Finally, the fact that long term memorability was better predicted using short term predictions indicates that we failed at capturing the memorability decay of a scene from a few minutes to a few days. In the future, we would like to focus more on this aspect of the task.

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] 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)*. 4724–4733.

[2] Mihai Gabriel Constantin, Bogdan Ionescu, Claire-Helene Demarty, Ngoc Q. K. Duong, Xavier Alameda-Pineda, and Mats Sjoberg. 2019. The Predicting Media Memorability Task at MediaEval 2019. *Proc. MediaEval workshop* (2019).

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805

[4] Huet B. Francis, D. 2019. L-STAP : Learned Spatio-Temporal Adaptive Pooling for Video Captioning. In *First International Workshop on AI for Smart TV Content Production (AI4TV)*.

[5] Rohit Gupta and Kush Motwani. 2018. Linear Models for Video Memorability Prediction Using Visual and Semantic Features. In *MediaEval*.

[6] 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)*. 770–778.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014).

[9] Yuncheng Li, Yale Song, Liangliang Cao, Joel R. Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. *CoRR* abs/1604.02748 (2016). http://arxiv.org/abs/1604.02748

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*.

[11] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *CoRR* abs/1703.03130 (2017). http://arxiv.org/abs/1703.03130

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

[13] 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*. Gaithersburg, MD, USA.

[14] Duy-Tue Tran-Van, Le-Vu Tran, and Minh-Triet Tran. 2018. Predicting Media Memorability Using Deep Features and Recurrent Network. In *MediaEval*.

[15] Jianxiong Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. 3485–3492.

[16] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5288–5296.

B.3 EURECOM ISWC 2019 conference paper [2]

This paper describes a method for automatically build and deploy Web APIs on top of a knowledge graph.

Easy Web API Development with SPARQL Transformer

Pasquale Lisena[1][0000−0003−3094−5585], Albert Meroño-Peñuela[2][0000−0003−4646−5842], Tobias Kuhn[2][0000−0002−1267−0234], and Raphaël Troncy[1][0000−0003−0457−1436]

[1] EURECOM, Sophia Antipolis, France
pasquale.lisena@eurecom.fr, raphael.troncy@eurecom.fr
[2] Vrije Universiteit, Amsterdam, The Netherlands
t.kuhn@vu.nl, albert.merono@vu.nl

Abstract. In a document-based world as the one of Web APIs, the triple-based output of SPARQL endpoints can be a barrier for developers who want to integrate Linked Data in their applications. A different JSON output can be obtained with SPARQL Transformer, which relies on a single JSON object for defining which data should be extracted from the endpoint and which shape should they assume. We propose a new approach that amounts to merge SPARQL bindings on the base of identifiers and the integration in the `grlc` API framework to create new bridges between the Web of Data and the Web of applications.

Keywords: SPARQL · JSON · JSON-LD · API

1 Introduction

The Semantic Web is a valuable resource of data and technologies, which is having a crucial role in realising the initial idea of Web. RDF can potentially represent any kind of knowledge, enabling reasoning, interlinking between datasets, and graph-based artificial intelligence. Nevertheless, a structural gap exists that is limiting a broader consumption of RDF data by the community of Web developers. Recent initiatives such as EasierRDF[3] are strongly pushing the proposal of new solutions for making Semantic data on the Web *developer friendly* [3, 10].

We focus here on the output format of SPARQL endpoints, and in particular, query results in the JSON format [24]. This standard is part of the SPARQL W3C recommendation [12], introduced with the purpose of easing the consumption of the data by Web (and non-Web) applications. The format consists of a set of all possible bindings (of the form `<variable, value>`) that satisfies the query. This is not handy for efficient processing by clients, which would prefer nested objects (document-based data structures) rather than this representation of triples (graph-oriented data structures). An example of this is shown in Figure 1.

[3] https://github.com/w3c/EasierRDF

Fig. 1. A SPARQL query (a) extracting a list of Italian cities with picture, label and belonging region, of which the URI and the Italian name are also requested. In the standard output of the endpoint (c), the city of Siena is represented by both object A and B, while the transformed output (b) offers a more compact structure.

Given this situation, we identify four tasks that developers have to fulfil:

1. Skip irrelevant metadata. A typical SPARQL output contains a lot of metadata that are often not useful for Web developers. This is the case of the `head` field, which contains the list of variables that one might find in the results. In practice, developers may ignore completely this part and check for the availability of a certain property directly in the JSON tree.

2. Reducing and parsing. The value of a property is always wrapped in an object with at least the attributes *type* (URI or literal) and *value*, containing the

information. As a consequence, this information is bounded at a deeper level in the JSON structure than the one the developer expects. In addition, each literal is expressed as a string value with a datatype, so that numbers and booleans need to be casted.
3. Merging. As the query results represent all the valid solutions of the query, it is possible that two bindings differ only by a single field. When the number of properties that have multiple values grows (i.e. multilingual names, multilingual descriptions, a set of images), the endpoint returns even more results, one for each combination of values. The consumption of such data requires often to identify all the bindings which represent a given entity, merging the objects on the URI. The presence of more variables on which the merging can be performed can further complicate the merging process.
4. Mapping. The Web developer may want to map the results to another structure – i.e. for using them as input to a library – or vocabulary such as *schema.org*.

In addition to this, the support for curating and reusing SPARQL queries is sub-optimal, these queries typically end up being hard-written in the application code. A specifically unsettling case of these Linked Data (LD) APIs, which refer to those APIs that just wrap underlying SPARQL functionality. To solve this problem, various works have provided bridges between the Web of Data and the developers. `grlc` is a software for the automatic generation of Web APIs from SPARQL queries contained in GitHub repositories [16]. **SPARQL Transformer**[4] is a library that gives a chosen structure to the SPARQL output. The library is able to perform all the above mentioned tasks, helping Web developers in the manipulation of data from the Web.

This paper largely extends [15] with a more organic description of the module, the integration of SPARQL Transformer in `grlc` and Tapas, a playground application for testing the query outcome and an evaluation on performance and usability. Moreover, the library has been ported to Python, and a set of new features have been included, most importantly the support of OFFSET (allowing pagination, e.g. in `grlc`) and language filtering for the management of multilanguage APIs. The remainder of this paper is structured as follows: we propose a thorough review of other works which aim to ease the consumption of RDF data and their limitations in Section 2. We introduce the new JSON format for queries in Section 3, which feeds the SPARQL Transformer library detailed in Section 4. The work is finally evaluated in Section 5, while some conclusions and future work are presented in Section 6.

2 Related Work

The need for overcoming the issues about the usage of SPARQL output in real-life applications has inspired different works. One of the first proposed solutions

[4] SPARQL Transformer is available at https://github.com/D2KLab/sparql-transformer as a JavaScript library, while a Python implementation is available at https://github.com/D2KLab/py-sparql-transformer.

consists in a strategy for representing the SPARQL output in a tabular structure, to address the creation of HTML reports [1].

Wikidata SDK [14] takes care of the reduction and parsing tasks through a precise function[5] that transforms the JSON output to a simplified version by reading the variable names. However this implementation does not address the problem of merging.

The conversion of RDF data can rely on the *SPARQL Template Transformation Language (STTL)* [4]. Those transformation templates (as strings) are exploited for shaping the results of the SPARQL query. Moreover, STTL exposes a significant number of functions, especially when combined with LDScript [5]. Among the limits of this approach is the absence of any support for converting the results to JSON-LD. No merging strategy is also studied in this approach.

The W3C RDFJS Community Group[6] is heavily contributing to the effort of offering a tool to JavaScript developers for using RDF data. The major outcome of the initiative is a low-level interface specification for the interoperability of RDF data in JavaScript environments [2]. RDFJS brings the graph-oriented model of RDF into the browser, allowing developers to directly manipulate triples.

The `CONSTRUCT` query format – included in the W3C SPARQL Specification [12] – can be seen as a way for mapping the SPARQL results into a chosen structure, following one of the standard SPARQL output formats, including JSON-LD. An attempt has been realised by the command-line library `sparql-to-jsonld` [17]. The need for three different inputs – a `SELECT` query, a `CONSTRUCT` or `DESCRIBE` query, and a JSON-LD frame – indirectly proves that a sole `CONSTRUCT` for shaping JSON with non predefined structure is not sufficient. Indeed, the `CONSTRUCT` keyword can only generate triplesets, from which the generation of JSON tree-like documents is ambiguous. This is inconvenient for developers, and leads to the problem of how to change the structure of the query result. JSON-LD Framing[7] overcomes this problem, but, in our opinion, the combination is not easier for developers who would have to write and keep in sync the two parts (query and result shape). The complexity of writing a `CONSTRUCT` query – i.e. with respect to a `SELECT` one – can be an additional deterrent for its usage. Furthermore, literals are not parsed and they are always represented as objects, and aggregate functions are not supported.

JSON Schema is a format for defining the structure of a JSON object. Although it is a powerful tool for validation – for example – of forms and APIs, there are no evident benefits for JSON reshaping purposes [29].

The development of *SOLID* framework for decentralised LD applications [28] gives popularity to its module *LDflex*[8] for retrieving and manipulating Linked Data. LDflex allows the user to browse nodes in the graph by accessing to JS

[5] https://github.com/maxlath/wikidata-sdk/blob/master/docs/simplify_sparql_results.md
[6] https://www.w3.org/community/rdfjs/
[7] https://www.w3.org/TR/json-ld11-framing/
[8] https://github.com/RubenVerborgh/LDflex

properties. Thus, the paradigm of this module is different, consisting in navigating the graph following the links, rather than finding solutions to structured queries.

There is abundant work in SPARQL query repositories, which are typically used to study the efficiency and reusability of querying. For example, in [21] authors use SPARQL query logs to study differences between human and machine executed queries; in [13], these logs are used to understand the semantic relations between queried entities. Saleem et al. [23] propose to "create a Linked Dataset describing the SPARQL queries issued to various public SPARQL endpoints".

There is also a large body of Semantic Web literature on Linked Data and Web Services [9, 20]. In [25] and the smartAPI [30], the authors propose to expose REST APIs as Linked Data, and enumerate the advantages of using Linked Data technology on top of Web services. In the opposite direction, the Linked Data API specification[9] and the W3C Linked Data Platform 1.0 specification, describe "the use of HTTP for accessing, updating, creating and deleting resources from servers that expose their resources as Linked Data"[10]. Our work follows this direction, and is more related to providing APIs that facilitate Linked Data access and query results consumption. The OpenPHACTS Discovery Platform for pharmacological data [11], LDtogo [19] and the BASIL server [6] use SPARQL as an underlying mechanism to implement APIs and provide Linked Data query results. Influenced by these works, `grlc` [16], a technology we extend in this paper, decouples query storage from API implementations by leveraging queries uniquely and globally identified by stable and de-referenceable URIs, automating the query construction process.

Recent works realised an interoperability between the GraphQL language[11] and RDF, performing in this way a conversion in JSON of the data in an endpoint [27]. The same syntax of GraphQL allows to produce a JSON object with different levels of nested nodes. Some of these solutions rely on automatic mappings of variables to property names (Stardog[12]), while others rely on a schema (HyperGraphQL[13]) or a context (GraphQL-LD [26]) which the developer is in charge to provide. None of those approaches implements any strategy for detecting and merging bindings referring to the same entity.

3 The JSON query syntax

As seen in the experiences reported in Section 2, the natural choice of format for defining and developing a transformation template involves JSON or its JSON-LD serialisation, which is usually added to the SPARQL query. The names of the variables used should match between the template and the query, making the developing process error-prone.

[9] https://github.com/UKGovLD/linked-data-api
[10] https://www.w3.org/TR/2015/REC-ldp-20150226/
[11] https://graphql.github.io/
[12] https://www.stardog.com/
[13] https://www.hypergraphql.org

```
{
  "proto": {
    "id" : "?id",
    "name": "$rdfs:label$required",
    "image": "$foaf:depiction",
    "region": {
      "id" : "$dbo:region$required",
      "name": "$rdfs:label$lang:it"
    }
  },
  "$where": [
    "?id a dbo:City",
    "?id dbo:country dbr:Italy"
  ],
  "$limit": 100
}
```

Listing 1.1. The JSON version of the SPARQL query in Figure 1

```
SELECT DISTINCT ?id ?v1 ?v2 ?v3r ?v31 WHERE {
    ?id a dbo:City.                          # 12
    ?id dbo:country dbr:Italy.               # 13
    ?id rdfs:label ?v1.                      # 4
    OPTIONAL { ?id foaf:depiction ?v2  }. # 5
    ?id dbo:region ?v3r .                    # 7
    OPTIONAL { ?v3r rdfs:label ?v31 .
        FILTER(lang(?v31) = "it") }          # 8
}
LIMIT 100                                    # 15
```

Listing 1.2. The intermediate SPARQL query. The comments contain line numbers which identify which part of the JSON query in Listing 1.1 generates the statement.

Our proposal is to use a single JSON object, called *JSON query*, with the double role of declaring how to find the information (query) and which structure is expected in its output (template). These properties put the JSON query at a certain distance also from SPARQL `CONSTRUCT`, in which the query and the final structure are two distinct parts of the query.

The syntax of JSON queries consists of two main parts (Listing 1.1):

- the prototype definition, which describes the output structure, expressed as an object and introduced by the `proto` property;
- a set of rules to be included in the SPARQL query, defined through a set of properties starting with the `$` sign, e.g. `$where` and `$limit`.

JSON queries can be expressed in two different formats, producing coherently the output: plain JSON and JSON-LD. The latter foresees a slightly different syntax in order to return an output compliant with the JSON-LD specification.

This version of the query allows to specify a JSON-LD context, and can be used for mapping the results into a chosen vocabulary. We refer to the documentation [14] for more details.

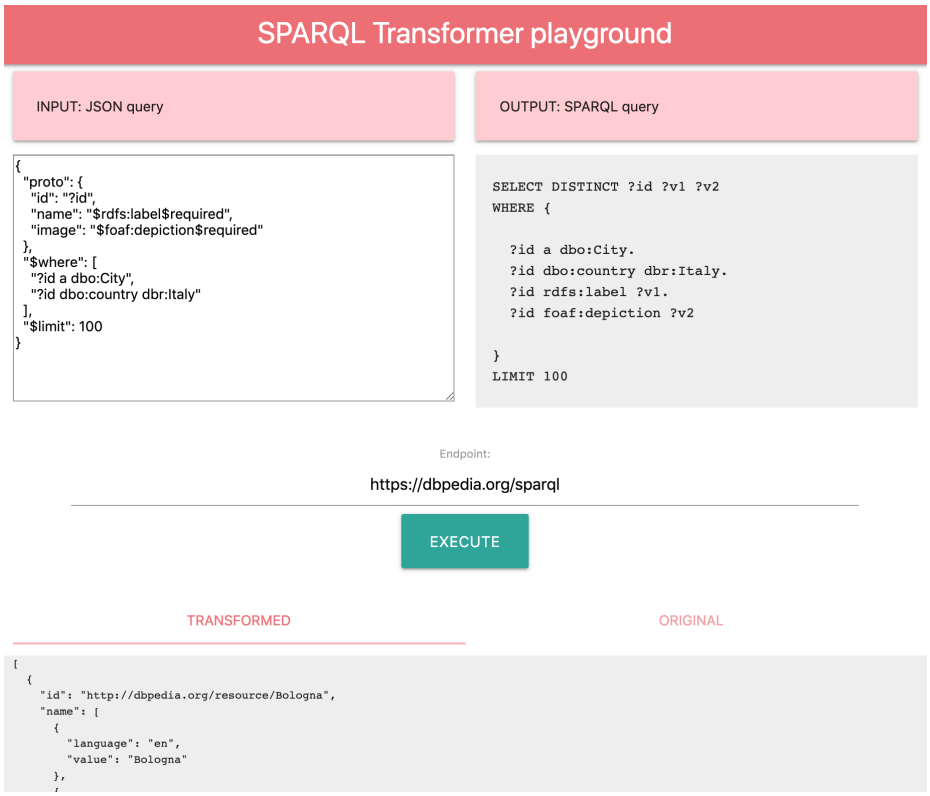

Fig. 2. User interface of SPARQL Transformer playground

A Web application called **SPARQL Transformer playground**[15] has been developed in order to quickly test JSON queries. The application is live converting the JSON into a corresponding SPARQL query, so that the user can appreciate every single change. In addition, it is possible to execute the query against a given endpoint, and the user interface offers the possibility of comparing the transformed output with the original one (Figure 2).

[14] https://github.com/D2KLab/sparql-transformer
[15] https://d2klab.github.io/sparql-transformer/

3.1 The prototype definition

By prototype, we mean the common structure each object in output should respect. It is designed as an ordinary JSON object, in which the leaf nodes will be replaced by incoming data according to specific rules. In particular:

1. **variable nodes**, which start with a question mark `"?"` (like `?id` or `?city`), are replaced by the value of the homonym SPARQL variable;
2. **predicate nodes**, which starts with a `"$"` sign, are replaced by the object of a specific RDF triple;
3. **literal nodes**, which cover all the other contents, are not replaced and will be present as is in the output, regardless of the query results.

In the transforming process, SPARQL triples will be automatically generated from the prototype. Referring to case 2, the following syntax is used:

```
$<SPARQL PREDICATE>[$modifier[:option]...]
```

The first parameter is the SPARQL predicate, which can be a property or a property path, e.g. `rdfs:label`, `foaf:depiction`, etc. This kind of node will be replaced by the object of an RDF triple having as predicate the one given inline. As subject, the variable of the sibling *merging anchor* is selected if it exists; otherwise, the closer merging anchor among the parent nodes. The merging anchors are all the fields in the JSON introduced with the `id` property. If this variable does not exist, it is set to `?id` by default. In other words, each level in the JSON tree may declare a specific subject through the merging anchor, which will be the subject of all the predicates in the scope. Listing 1.1 includes two merging anchors at line 3 and 7: the former acts as subject of the name, image, and region; while the region name refers to the latter.

The role of the *merging anchor* is crucial for the following steps. In fact, two result objects having the same id will be considered as the same item and their properties will be merged. This will happen at each level of the JSON tree. This controlled way of aggregating SPARQL results ensures a more compact while not less informative output, ready to be used by Web developers.

Both variable and predicate nodes can accept some modifiers appended at the end of the string, separated by the `$` sign. These elements are taken in account when writing the SPARQL query. For example, `$required` avoids the predicate to be considered optional (the default behaviour), while `$var` assigns a specific SPARQL variable as object (e.g. `$var:?myVar`), so that it can be addressed in other modifiers. Other possibilities include filtering by language (`$lang:it` or `$bestlang:en;q=1, it;q=0.7 *;q=0.1`) or sample those values (`$sample`).

3.2 The root $-properties

A set of `$`-properties give access to the SPARQL features indicated by their name (`$limit`, `$groupby`, etc). These properties are directly assigned to the root of the JSON query object, and will not appear in the final output. Among them, some additional `WHERE` clauses – in the triple format – can be declared in the

`$where` field. The `$lang` modifiers set the language chosen for all the `$bestlang` in the prototype. An exhaustive list of implemented `$`-properties is reported in Table 1.

Table 1. Supported root `$`-properties

PROPERTY	INPUT	DESCRIPTION
`$where`	string, array	Add where clause in the triple format.
`$values`	object	Set VALUES for specified variables as a map.
`$limit`	number	LIMIT the SPARQL results
`$distinct`	boolean	Set the DISTINCT in the select (default `true`)
`$offset`	number	OFFSET applied to the SPARQL results
`$orderby`	string, array	Build an ORDER BY on the variables in the input.
`$groupby`	string, array	Build an ORDER BY on the variables in the input.
`$having`	string, array	Allows to declare the content of HAVING.
`$filter`	string, array	Add the content as a FILTER.
`$prefixes`	object	Set the prefixes in the format `"prefix": "uri"`.
`$lang`	string	Default language in the Accept-Language standard. [8]

4 Implementation

The implementation of SPARQL Transformer relies on three main blocks, each one having a specific function (Figure 3).

The **Parser** reads the input JSON query and parses its content. The prototype is extracted and a SPARQL variable – which here acts as a placeholder – is assigned to all the predicate nodes. Contextually, the SPARQL `SELECT` query (Listing 1.2) is generated: the predicate nodes are translated into `WHERE` clauses according to the rules defined in Section 3.1 and taking into account the modifiers. The root `$`-properties are parsed and inserted in the query, which is then passed to the **Query Performer**. This module is in charge of performing the request to the SPARQL endpoint and returning the results in the SPARQL JSON output format. The query performer can be replaced by the user with a custom one, for fulfilling different requirements for accessing the endpoint (e.g. authentication) or for integration into more complex environments (as done during the integration with `grlc`).

Finally, the **Shaper** accesses the results, discarding the side information included in the `head` field and directly accessing the bindings. The latter ones are applied to the prototype in sequence, matching the SPARQL variables to the placeholders separately for each binding. In this phase, the data-type of the binding is checked, eventually parsing the value to Boolean, integer or float. When a result binding does not contain a certain value – which happens when the variable is `OPTIONAL` –, the property is removed from the instance. Then, the instances which have a common value for the merging anchor are identified

and their properties are compared, in order to keep all the distinct values without repetition. Recursively, the same merging strategy is applied to the nested objects. Finally, they are serialised in JSON and returned as output.

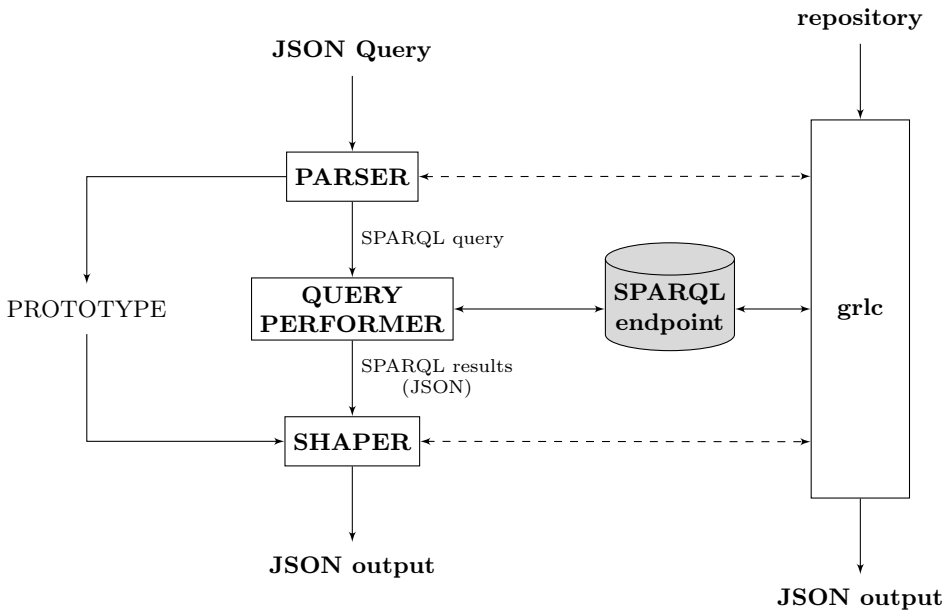

Fig. 3. The application schema of SPARQL Transformer

The SPARQL Transformer library is available in two different implementations in JavaScript and Python, published respectively on the NPM Package Manager[16] and the Python Package Index[17] (PyPI). The JavaScript version has been recently converted in an ECMAScript Module [7] and it is designed to both work in Node.js and in the browser. The Python version return a `dict` object, which can be directly manipulated by a script or serialised in JSON.

Since version 1.3, SPARQL Transformer is included in the `grlc`[18] framework, which is now able to generate Web APIs from the JSON queries contained in a given GitHub repository. The integration involved the Parser and the Shaper: the former is executed before each access to the SPARQL query, keeping in memory the prototype for being shaped once SPARQL results are back. The JSON query file can include the configuration options for `grlc` in an homonym field. For maximising the compatibility, the options can be specified as a YAML string or in JSON. The support to JSON queries includes all the features of

[16] https://www.npmjs.com/package/sparql-transformer
[17] https://pypi.org/project/SPARQLTransformer/
[18] http://grlc.io/

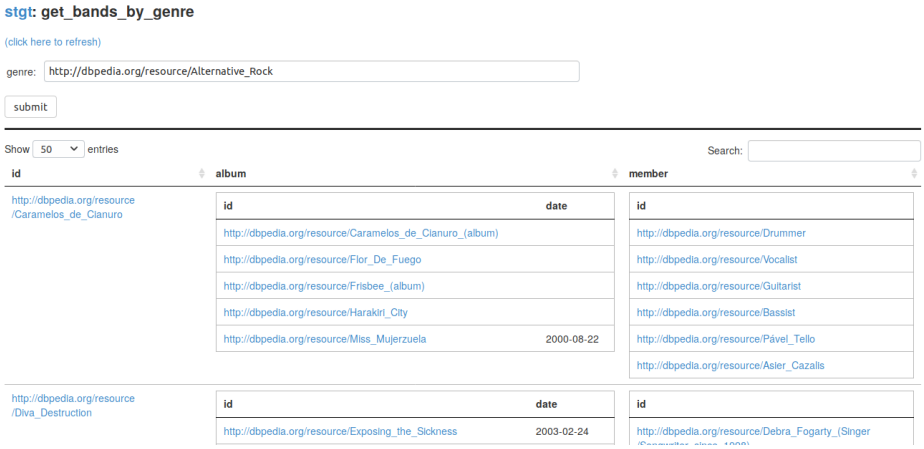

Fig. 4. Screenshot of the Tapas interface

`grlc`, such as the pagination and the selection of query parameters. In addition, a `lang` query parameter can change the value of the `$lang` property of the query, allowing the development of multi-language APIs. Further development involved the upgrade of `grlc` to the latest Python version.

Moreover, SPARQL Transformer queries are now also supported by Tapas[19]. Tapas is a small interface module implemented in HTML and JavaScript that reads the specification of an instance of a `grlc` API and turns it into a nice and simple HTML interface. The elements of the API specification are in a straightforward manner transformed into HTML form elements, which the user can fill in to access the service by pressing the *submit* button. Tapas asynchronously calls the API via `grlc` and shows the results at the bottom part of the same page using the YASR component of the YASGUI interface [22] to display the SPARQL query results in a user-friendly manner.We extended Tapas to also support SPARQL Transformer queries and display the results in an equally user-friendly manner. Unlike the flat tables produced by YASR for the common kind of SPARQL results, the nested results of a SPARQL Transformer query are shown as nested tables in Tapas. An example of this can be seen in Figure 4, showing a screenshot of the query interface and its results for an exemplary SPARQL Transformer query about music bands, with the nested tables derived from the nested structure of the SPARQL Transformer results. Tapas together with `grlc` thereby allow us to automatically generate an intuitive interface for technically-minded end users just from the query file in a completely general and generic manner.

[19] https://github.com/peta-pico/tapas

5 Evaluation

As evidence of *current* use, we have deployed this tool in two communities driven by H2020 projects which have adopted both SPARQL Transformer and `grlc`. MeMAD[20] uses it to generate automatically an API on top of a knowledge graph describing TV and radio programs which are also automatically annotated. The resulting semantic metadata is hence integrated in the professional Media Asset Management system Flow developed by Limecraft. SILKNOW[21] uses it to generate an API on top of a knowledge graph describing silk-related objects from 10 museums. The generated API is used to empower an exploratory search engine and a virtual assistant.

To provide evidence of *prospective* use of our approach, we carried out two kinds of evaluations:

- an experiment for measuring the compactness of the results and the execution time of SPARQL Transformer;
- a user survey on the preference of users on using a system that presents Linked Data query results through SPARQL Transformer, versus another that does so through traditional SPARQL results rendering.

5.1 Quantitative evaluation

We test the Python implementation of SPARQL Transformer on a set of five queries detailed in the DBpedia wiki[22] in order to ensure a certain generality. The set involves different SPARQL features (filters, ORDER BY, language filtering, optional triples). Those SELECT queries have been manually converted into JSON queries — with 1 or 2 levels of objects in the JSON tree —, making sure that the transformed query was equal to the original one (variable names apart).

Each query has been resolved against a local instance of the English DBpedia[23], with a traditional SPARQL client for the SPARQL queries and with SPARQL Transformer for the JSON queries. Each execution has been repeated 100 times, with a waiting time of 5 seconds between consecutive executions, in order to obtain an average result as much as possible not correlated to any workload of the machine.

The results in Table 2 shows that the average execution time of SPARQL Transformer is slightly higher with respect to normal SPARQL queries, never surpassing 0.1 seconds (limit of the instantaneous feeling according to [18]). The difference in percentage, computed as $100 * (t_{sparql} - t_{json})/avg(t_{sparql}, t_{json})$, do not reveal any regularity in the time increment, even if some patterns suggest that it depends on the number of results and variables for each result. The same dimensions seem to impact also the gap in number of results, smaller in the JSON

[20] https://memad.eu/
[21] http://silknow.eu/
[22] https://wiki.dbpedia.org/onlineaccess, Section 1.5
[23] The setup of the endpoint on a local machine relied on *Dockerized-DBpedia*, available at https://github.com/dbpedia/Dockerized-DBpedia

query responses because of the merging strategy. It is interesting to point out that such difference exists between all valid combinations of values for requested variables and the number of real-world object described. This is evident in the first query, about people born in Berlin, in which the combinations of names in different languages and birth or death date in different formats almost double the number of results. As a consequence, the Prince Adalbert of Prussia[24] appears in 8 distinct (and even non-consecutive) bindings because of its four names and two versions of its death date, correctly merged in the more compact transformed version. The experiment is further detailed in the GitHub repository[25].

Table 2. Differences in number of results and execution time between SPARQL and JSON queries. For each query, is also reported the number of requested variables.

		N. RESULTS			TIME (ms)			
QUERY NAME	**N. VAR**	**json**	**sparql**	**diff %**	**json**	**sparql**	**diff**	**diff %**
1. Born in Berlin	4	573	1132	49%	168	101	67	50%
2. German musicians	4	257	290	11%	61	49	12	22%
3. Musicians born in Berlin	4	109	172	37%	59	51	8	14%
4. Soccer players	5	70	78	10%	210	203	7	3.7%
5. Games	2	981	1020	4%	121	70	51	54%

5.2 User Survey

In order to evaluate the usefulness of the query results as presented by SPARQL Transformer to potential (technically-minded) end-users and developers and to compare them to a more traditional, table-centric provision of SPARQL query results, we conducted a user survey. We hypothesized that the level of nesting would play an important role, as classical SPARQL results are flat tables whereas the JSON structure of SPARQL Transformer allows for nesting.

We therefore constructed a pair of queries in SPARQL Transformer syntax and its corresponding plain SPARQL version for each of three levels of nesting: no nesting (Level 0), one nested structure (Level 1), and two nested structures (Level 2). These queries are all about bands and their albums and members, and they can be run through the DBpedia SPARQL endpoint. An example of two nested structures as found in Level 2 can be seen in Figure 4 (the two nested structures being *album* and *member*). We then ran each of these six queries and stored the resulting JSON files (i.e. the files generated by SPARQL Transformer and the standard JSON files with the original SPARQL results, respectively). Moreover, we also ran these on Tapas to compare the user interface aspects that come with the different representations and nesting styles, and we made

[24] http://dbpedia.org/resource/Prince_Adalbert_of_Prussia_(1811-1873)
[25] A notebook is available at https://github.com/D2KLab/py-sparql-transformer/blob/master/evaluation/test.ipynb

Table 3. The results of the user survey

		preference for our system							
Type	Level	−2	−1	0	1	2	avg.	p-value	
JSON results	0 (no nesting)	6	6	4	13	26	0.85	0.0001980	*
	1 (one nesting)	5	5	3	21	21	0.87	0.000009063	*
	2 (two nestings)	3	9	5	17	21	0.80	0.0003059	*
Tapas interface	0 (no nesting)	4	8	3	19	21	0.82	0.0001275	*
	1 (one nesting)	3	10	2	20	20	0.80	0.0002685	*
	2 (two nestings)	4	7	3	16	25	0.93	0.00003589	*

screenshots of the result tables. All these files, including queries, their results, and the Tapas screenshots, can be found online[26].

Based on these query results and screenshots, we then created a questionnaire, where we asked the participants for each of the six cases (JSON files and screenshots for each of the three nesting levels) whether they preferred SPARQL Transformer (referred to as "System A") or the classical SPARQL output (referred to as "System B"), displayed using the YASR component of YASGUI. The possible answers consisted of the five options *Strongly prefer B* (value -2), *Slightly prefer B* (-1), *Indifferent* (0), *Slightly prefer A* (1), and *Strongly prefer A* (2). We also asked the participants whether they consider themselves primarily researchers, developers, or none of these two categories, and we asked about their level of expertise with SPARQL and JSON. The questionnaire can be found online[27].

We then asked people to anonymously participate in this user survey via Linked Data related mailing lists (W3C SemWeb list), and internal group lists of Semantic Web groups at VU Amsterdam and EURECOM, in addition to the SIKS list addressing Dutch universities. The form was accessible for 5 days. In this way, we got responses from 55 participants (40 researchers, 9 developers, 6 others). Their level of expertise on SPARQL and JSON was mixed, with average values of 2.44 and 2.87, respectively, on a scale from 0 to 4. Eight participants had no knowledge of SPARQL at all, while only one participant had no knowledge of JSON.

Table 3 shows the results of the survey (the full table can also be found online[28]). We see that we got the full range of replies for all questions, but also that a clear majority prefers our system slightly (1) or even strongly (2). The average values for both types (JSON and Tapas) and all three nesting levels are between 0.80 and 0.93, i.e. close to the value that stands for a slight preference of our system (1) and clearly above the value that stands for an indifference between the two (0).

[26] https://github.com/tkuhn/stgt/
[27] https://github.com/tkuhn/stgt/blob/master/eval/questionnaire-form.md
[28] https://github.com/tkuhn/stgt/raw/master/eval-results/questionnaire-results.ods

To test whether the preference towards our system is statistically significant, we used a sign test in the form of a binomial test on the answers that were positive (preference of our system) or negative (preference of the existing system), excluding the zero cases (indifference). This test, therefore, does not take the distinction between slight and strong preference into account, but only which system was preferred. The final column of Table 3 lists the p-values of this test, showing that the effect is highly significant for all six cases.

The results, however, do not support our hypothesis that the level of nesting has an effect on the preference for our system. Throughout all nesting levels, the users expressed clear and significant preference for our system, but this preference did not increase with increased nesting levels.

6 Conclusion and Future Work

SPARQL Transformer offers to Web developers a different way of approaching RDF datasets. The adoption of a novel JSON format for defining both the query and the template makes it possible to realise self-contained files. When collected in a GitHub repository, these files can be easily transformed into Web APIs with `grlc`, completing the decoupling between query, post-processing and consumption in the application, and query results can moreover be presented in a simple and user-friendly manner via Tapas. The evaluation reveals that the restructuring and merging pipeline of SPARQL Transformer has an important impact in making the SPARQL results more usable and understandable by humans.

Differently from other works, SPARQL Transformer allows developers to use one single file for querying and mapping, and even with some limits – i.e. not being as expressive as SPARQL – can be of benefit for fast prototyping of web application.

Further development can improve SPARQL Transformer in order to fulfil a wider range of needs. The query support can be extended to other SPARQL operations, like ASK, INSERT and DELETE, going towards the realisation of full REST APIs on top of SPARQL endpoints. Aggregate functions (e.g. COUNT, SUM) should join the set of available features in the near future. We will further investigate the use of JSON frames, in order to extract the Shaper component from the library and make it available for standalone use.

Currently, the JSON syntax does not foresee any standard way for representing dates, which are therefore represented as plain strings. Alternative representations for dates should be found taking into account developer requirements, even listening and involving them in the final decision. Possibly, the solution should also involve other related data-types, like `xsd:gYear` or `xsd:duration`.

We plan to run another evaluation of this work, this time focused on the creation scenario, consisting in an interview on query writing with SPARQL Transformer and on API management with `grlc`.

Finally, we are currently planning to offer more customisation possibilities to users. Some examples include the choice of a different merging anchor (currently forced to `id` or `@id`); the possibility of ignoring language tags in the results

(avoiding the presence of a language-value object); and the chance of distinguishing between IRIs (as resource references) and IRIs in lexical forms.

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 CLARIAH project of the Dutch Science Foundation (NWO). We want to thank Ilaria Tiddi for her support and suggestions on combining our work.

References

1. Abburu, S., Babu, G.S.: Format SPARQL Query Results into HTML Report. International Journal of Advanced Computer Science and Applications (IJACSA) **4**(6), 144–148 (2013)
2. Bergwinkl, T., Luggen, M., elf Pavlik, Regalia, B., Savastano, P., Verborgh, R.: Interface Specification: RDF Representation, Draft Report. Tech. rep., W3C (2017)
3. Booth, D., Chute, C.G., Glaser, H., Solbrig, H.: Toward Easier RDF. In: W3C Workshop on Web Standardization for Graph Data. Berlin, Germany (2019)
4. Corby, O., Faron-Zucker, C., Gandon, F.: A generic RDF transformation software and its application to an online translation service for common languages of linked data. In: 14^{th} International Semantic Web Conference (ISWC). pp. 150–165. Bethlehem, Pennsylvania, USA (2015)
5. Corby, O., Faron-Zucker, C., Gandon, F.: LDScript: a Linked Data Script Language. In: 16^{th} International Semantic Web Conference (ISWC). pp. 208–224. Vienna, Austria (2017)
6. Daga, E., Panziera, L., Pedrinaci, C.: A BASILar Approach for Building Web APIs on top of SPARQL Endpoints. In: International Workhop on Services and Applications over Linked APIs and Data (SALAD). vol. 1359. CEUR Workshop Proceedings, Bethlehem, Pennsylvania, USA (2015)
7. Ecma International: ECMAScript 2015 Language Specification. 6^{th} Edition. ECMA-262. Tech. rep., Ecma International (2015)
8. Fielding, R., Gettys, J., Mogul, J.C., Frystyk Nielsen, H., Masinter, L., Leach, P.J., Berners-Lee, T.: Hypertext transfer protocol (HTTP/1.1): Header Field Definitions. RFC 2616. Tech. rep., Internet Engineering Task Force (2014)
9. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures (2000), PhD Thesis
10. Gandon, F., Michel, F., Corby, O., Buffa, M., Tettamanzi, A., Faron Zucker, C., Giboin, A., Cabrio, E., Villata, S.: Graph Data on the Web: extend the pivot don't reinvent the wheel. In: W3C Workshop on Web Standardization for Graph Data. Berlin, Germany (2019)
11. Groth, P., Loizou, A., Gray, A.J., Goble, C., Harland, L., Pettifer, S.: API-centric Linked Data integration: The Open PHACTS Discovery Platform case study. Web Semantics: Science, Services and Agents on the World Wide Web **29**(0), 12 – 18 (2014)
12. Harris, S., Seaborne, A.: SPARQL 1.1 query language – W3C recommendation. Tech. rep., W3C (2013)

13. Huelss, J., Paulheim, H.: What SPARQL Query Logs Tell and Do Not Tell About Semantic Relatedness in LOD. In: Workshop on Negative or Inconclusive Results in Semantic Web (NoISE). pp. 297–308. Portoroz, Slovenia (2015)
14. Lathuilière, M.: `Wikidata SDK`. https://github.com/maxlath/wikidata-sdk (2015)
15. Lisena, P., Troncy, R.: Transforming the JSON Output of SPARQL Queries for Linked Data Clients. In: International Conference Companion on World Wide Web (WWW Companion). pp. 775–780. International World Wide Web Conferences Steering Committee, Lyon, France (2018). https://doi.org/10.1145/3184558.3188739, https://doi.org/10.1145/3184558.3188739
16. Meroño-Peñuela, A., Hoekstra, R.: grlc Makes GitHub Taste Like Linked Data APIs. In: The Semantic Web – ESWC 2016 Satellite Events. pp. 342–353. Heraklion, Greece (2016)
17. Mynarz, J.: `sparql-to-jsonld`. https://github.com/jindrichmynarz/sparql-to-jsonld (2016)
18. Nielsen, J.: Usability engineering. Elsevier (1994)
19. Ockeloen, N., de Boer, V., Aroyo, L.: LDtogo: A Data Querying and Mapping Framework for Linked Data Applications. In: The Semantic Web: ESWC 2013 Satellite Events. pp. 199–203. Montpellier, France (2013)
20. Pedrinaci, C., Domingue, J.: Toward the Next Wave of Services: Linked Services for the Web of Data. Journal of Universal Computer Science **16**(13), 1694—1719 (2010)
21. Rietveld, L., Hoekstra, R.: Man vs. Machine: Differences in SPARQL Queries. In: 4^th Workshop on Usage Analysis and the Web of Data (USEWOD). Anissaras, Greece (2014)
22. Rietveld, L., Hoekstra, R.: The YASGUI family of SPARQL clients. Semantic Web **8**(3), 373–383 (2017)
23. Saleem, M., Intizar Ali, M., Mehmood, Q., Hogan, A., Ngonga Ngomo, A.C.: LSQ: Linked SPARQL Queries Dataset. In: 14^{th} International Semantic Web Conference (ISWC). pp. 261–269. Bethlehem, Pennsylvania, USA (2015)
24. Seaborne, A.: SPARQL 1.1 query results JSON format – W3C recommendation. Tech. rep., W3C (2013)
25. Speiser, S., Harth, A.: Integrating Linked Data and Services with Linked Data Services. In: 8^{th} Extended Semantic Web Conference (ESWC). pp. 170—184. Heraklion, Greece (2011)
26. Taelman, R., Vander Sande, M., Verborgh, R.: GraphQLLD: Linked Data Querying with GraphQL. In: 17^{th} International Semantic Web Conference (ISWC), Poster & Demo Track. Monterey, California, USA (2018)
27. Taelman, R., Vander Sande, M., Verborgh, R.: Bridges between GraphQL and RDF. In: W3C Workshop on Web Standardization for Graph Data. Berlin, Germany (2019)
28. Verborgh, R.: Decentralizing the semantic web through incentivized collaboration. In: 17^{th} International Semantic Web Conference (ISWC), Blue Sky Track. vol. 2189 (Oct 2018)
29. Wright, A., Andrews, H.: JSON Schema: A Media Type for Describing JSON Documents. Tech. rep., Internet Engineering Task Force (2017), https://datatracker.ietf.org/doc/draft-handrews-json-schema/
30. Zaveri, A., Dastgheib, S., Whetzel, T., Verborgh, R., Avillach, P., Korodi, G., Terryn, R., Jagodnik, K., Assis, P., Wu, C., Dumontier, M.: smartAPI: Towards a more intelligent network of Web APIs. In: 14^{th} Extended Semantic Web Conference (ESWC). Portoroz, Slovenia (2017)

B.4 Aalto's ACM AI4TV 2020 paper

This paper describes the NER model that Aalto developed and submitted on the 2nd International Workshop on AI for Smart TV Content Production, Access and Delivery (ACM AI4TV 2020).

Named Entity Recognition for Spoken Finnish

Dejan Porjazovski
Aalto University
Espoo, Finland
dejan.porjazovski@aalto.fi

Juho Leinonen
Aalto University
Espoo, Finland
juho.leinonen@aalto.fi

Mikko Kurimo
Aalto University
Espoo, Finland
mikko.kurimo@aalto.fi

ABSTRACT

In this paper we present a Bidirectional LSTM neural network with a Conditional Random Field layer on top, which utilizes word, character and morph embeddings in order to perform named entity recognition on various Finnish datasets. To overcome the lack of annotated training corpora that arises when dealing with low-resource languages like Finnish, we tried a knowledge transfer technique to transfer tags from Estonian dataset. On the human annotated in-domain Digitoday dataset, out system achieved F1 score of 84.73. On the out-of-domain Wikipedia set we got F1 score of 67.66. In order to see how well the system performs on speech data, we used two datasets containing automatic speech recognition outputs. Since we do not have true labels for those datasets, we used a rule-based system to annotate them and used those annotations as reference labels. On the first dataset which contains Finnish parliament sessions we obtained F1 score of 42.09 and on the second one which contains talks from Yle Pressiklubi we obtained F1 score of 74.54.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Natural language processing**;

KEYWORDS

named entity recognition, speech recognition, low-resource

ACM Reference Format:
Dejan Porjazovski, Juho Leinonen, and Mikko Kurimo. 2020. Named Entity Recognition for Spoken Finnish. In *2nd International Workshop on AI for Smart TV Content Production: Affiliation; Access and Delivery (AI4TV'20), October 12, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3422839.3423066

1 INTRODUCTION

Named entity recognition (NER) is a natural language processing (NLP) task, in which the system aims to find entities in a text and classify them to predefined categories. The categories can vary based on the domain in which they are going to be used but some of the most common categories include: person, organization, product, location and date. NER is an integral part in larger areas such as information retrieval, question answering, machine translation and text summarization. It is a difficult problem because in most languages there is little annotated data available, especially in specific domains such as: chemistry, biology and medical fields. Entity ambiguity is another challenge that the system needs to deal with. For example "Facebook" can refer to both company and product, depending on the context it appears in.

In the past, researchers relied on hand-crafted features and gazetteers for solving this task, which requires in-domain knowledge [3, 9]. With the rise of machine learning (ML), researchers have tried different ML techniques in order to solve this task. The most popular approach is using Conditional Random Fields (CRFs) [11] which has been successfully applied to various NER tasks [14, 18].

With the increase of the computational power, deep neural networks have become more appealing, especially because they help to alleviate the need of domain experts and hand-crafted features. Recurrent neural networks, especially the Long Short-Term Memory (LSTM) [7] have been well suited for sequential tasks due to their ability to store information about sequences. Deep neural network architectures have been successfully applied in various NER tasks where they outperform CRF based models [1]. These systems used word embeddings as input features to the network. In 2015, a new neural network approach was proposed, which uses a CRF layer on top of a Bidirectional LSTM (BiLSTM). This model outperformed the previous neural network architectures and achieved the state-of-the-art results on the CoNLL 2013 dataset [8].

Although these approaches work well for most languages, frequent occurrences of inflections, derivations and compounding in morphologically rich languages makes their vocabulary large and increases the number of out-of-vocabulary (OOV) words. In order to overcome that issue, subword units such as characters or morphs have been proposed to replace words [6]. Character-level LSTM models have been applied on various languages and have been shown to give competitive results [10]. Furthermore, combining word and character representations has given an improvement over the existing models [13, 17]. Segmenting the words into morphs has been shown to improve the performance of the language models and reduce the out-of-vocabulary words by constructing the unseen words from morphs [2, 19].

Large vocabulary is not the only issue that arises when dealing with Finnish language. Lack of annotated data puts the Finnish language in a low-resource category. This constraint causes difficulties for training a NER system, especially when the annotated data is in a specific domain. In an attempt to overcome this limitation, different knowledge-transfer techniques have been proposed, which try to transfer tags from the source to the target language, in order to enrich the annotated corpora. [4, 23].

Another challenge that arises when doing NER is when we are dealing with unstructured data, such as an output of an automatic

AI4TV'20, October 12, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8146-8/20/10…$15.00
https://doi.org/10.1145/3422839.3423066

Table 1: Class distribution in Digitoday and Wikipedia datasets.

Class	Count Digitoday	Count Wikipedia
ORG	15445	1821
LOC	4159	1427
PER	6517	2492
DATE	3685	1862
PRO	11655	2135
EVENT	569	362
TOTAL	42030	10099

Table 2: Class distribution in Parliament and Yle Pressiklubi datasets.

Class	Parliament	Yle Pressiklubi
PER	104	1350
LOC	54	601
ORG	/	327
TOTAL	158	2278

speech recognition (ASR) system. Named entities are often capitalized, so the system relies on the capitalization in order to detect the entities, which causes problems for ASR output, where capitalization is neglected.

In this paper we propose a bidirectional LSTM-CRF architecture that utilizes words, characters and morphs in order to achieve competitive results in NER for Finnish language. Moreover, we are going to explore different ways of improving the performance of the system on ASR output. In order to deal with the low-resource limitations, we experimented with knowledge transfer from Estonian language using multilingual word embeddings for Finnish and Estonian languages, aligned in a single vector space.

2 DATA

We used the Digitoday dataset to train the model. The dataset was collected and provided by [16]. It consists of online Finnish technological news articles. There are 953 articles and 193,742 word tokens in the dataset. Since the articles are from one domain, the authors also provided a Wikipedia test for evaluating the system on out-of-domain data. Both datasets are annotated using the BIO annotation scheme [15]. The Wikipedia test set consists of 83 articles and 49,752 word tokens. The dataset consists of 6 named entity classes:

- PERSON (PER)
- LOCATION (LOC)
- ORGANIZATION (ORG)
- PRODUCT (PROD)
- EVENT (EVENT)
- DATE (DATE)

The class distribution of Digitoday and Wikipedia datasets is presented in Table 1. Both datasets provide top-level and nested-level entities. In our experiments we used only the top-level entities.

In order to test how well the system performs in an ASR setting, we used two datasets of ASR outputs. The first one contains Finnish parliament sessions and the second one contains talks from Yle Pressiklubi television show. Using a commercial rule-based system,

Table 3: Class distribution in Estonian dataset.

Class	Count
PER	12154
LOC	3508
ORG	9424
TOTAL	25086

we managed to obtain two NER tags for the parliament sessions and three tags for the Pressiklubi dataset. The Parliament dataset is in lowercase and without punctuation, whereas the Yle Pressiklubi dataset is re-capitalized. The class distribution for the ASR datasets is presented in Table 2. In Table 3 we can see the class distribution of the Estonian dataset that we used to transfer tags to Finnish and make the dataset less domain biased.

3 METHODS

In this section we present our architecture for NER, which utilizes word, character and morph representations. For agglutinative languages like Finnish, which have a rich vocabulary, the number of OOV words increases, which has an impact on the performance of the model. In order to mitigate this, besides the standard word embeddings, we augmented our model with morph and character representations of the words. To obtain the morphs, we used the Morfessor toolkit [20]. The architecture is depicted in Figure 1.

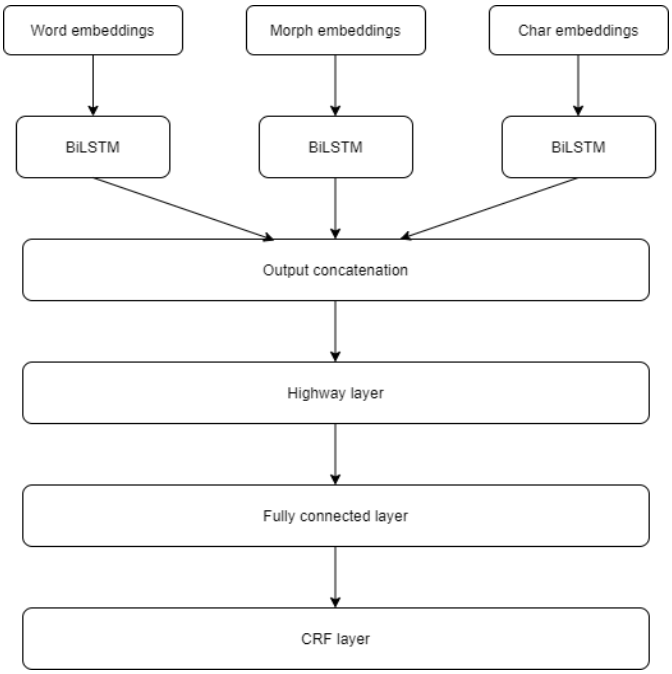

Figure 1: BiLSTM-CRF model that utilizes char, morph and word embeddings.

As we can see from the figure, word, character and morph embeddings are processed through separate BiLSTMs. The outputs of the BiLSTMs are concatenated in order to get a single representation. The concatenated outputs then go through a highway layer, which

is followed by a fully connected layer. At the end, the output of the fully connected layer goes through a CRF layer, which produces tag probabilities.

4 EXPERIMENTS

As described in section 2, we use the Digitoday dataset which contains technological articles, as well as the Wikipedia test set and the ASR outputs for testing the system on out-of-domain data. For the Digitoday and Wikipedia evaluations, we trained our system using the Digitoday train set and for the ASR evaluations we used the whole Digitoday dataset along with the Wikipedia test set for training.

In order to make a distinction between the first and the last word in a sentence and the rest of the words, we added "<*start*>" and "<*end*>" tokens to each sentence. For the morph-based subword modeling, we added boundary markers to enforce restrictions on the generated output. Different ways of adding markers enforce different restrictions. Some common types of markers are: "<*w*>", "<*m+*>", "<*+m*>", "<*+m+*>". In our experiments we used the "<*+m+*>" style marker since it has been shown to give best results for Finnish language modeling in ASR [21]. For example, the word 'mobiilikäyttöjärjestelmä' would be segmented as 'mobiili+ +käyttö+ +järjestelmä'.

The architecture has 2 BiLSTM layers and 4 highway layers. The embedding dimensions of words, chars and morphs are 300, 100, 100 respectively for the BiLSTM networks. The hidden sizes are 300, 75, 75 for words, chars and morphs. A dropout of 0.5 is added to the final BiLSTM outputs and 0.2 for each layer except for the last. After the highway layer, we added a dropout probability of 0.7. For training the model we used a batch size of 128 and RAdam optimizer [12] with learning rate of 0.001. All of the hyperparameters were chosen based on internal experiments that we did on the development set.

As baseline models we used a rule based system called FiNER and a neural network architecture called GÜNGÖR-NN [16]. The GÜNGÖR-NN architecture is described in more detail in [5]. In order to see the system's performance on ASR output, we used Finnish parliament sessions as well as the Yle Pressiklubi television show, which were decoded using a commercial ASR system. The datasets were annotated by named entity tags given by a rule-based system. We used those tags as the reference labels.

Doing NER on an ASR output has many challenges, such as recognition errors and missing capitalization and punctuation. When evaluating the model on the Parliament dataset, we decided to remove capitalization and punctuation from the training data, so that the system would learn in the ASR setting better.

Another issue that we faced was the out-of-domain problem, just like when testing our system on the Wikipedia dataset. To alleviate that problem we used knowledge transfer technique as described in the previous section. Because some of the tag translations were not very accurate, we used thresholding to keep only the translations that have high nearest neighbor candidate score in the target language. We did multiple experiments on the Digitoday dev set and found that a threshold value of 0.6 yields best results. Since person names and location names are almost the same in Finnish and Estonian, we kept them as they are in the Estonian and just transferred them to Finnish. This approach gave us an improvement over translating them as we did with the other entities.

Table 4: Overall micro average precision, recall and F1 scores for the top-level entities of Digitoday test set.

architecture	precision	recall	F1
FiNER	90.41	83.51	86.82
GÜNGÖR-NN	83.59	85.62	84.59
word+char+morph-LSTM	85.52	83.74	84.62
word+char+morph-LSTM+transfer	85.27	84.19	84.73

Table 5: Overall micro average precision, recall and F1 scores for the top-level entities of Wikipedia test set.

architecture	precision	recall	F1
FiNER	85.17	72.47	78.31
GÜNGÖR-NN	62.98	55.89	59.22
word+char+morph-LSTM	71.34	56.38	62.98
word+char+morph-LSTM+transfer	74.55	61.93	67.66

Table 6: Overall micro average precision, recall and F1 scores for the Parliament and Yle Pressiklubi datasets.

	Parliament data			Yle Pressiklubi data		
TAG	precision	recall	F1	precision	recall	F1
PER	46.11	89.25	60.81	80.00	85.71	82.76
LOC	14.53	69.39	24.03	76.92	86.96	81.63
ORG	/	/	/	55.56	26.79	36.14
avg	28.26	82.39	42.09	76.25	72.91	74.54

5 RESULTS

In this section we present the results obtained from the proposed BiLSTM-CRF architecture and compare them with the rule-based and neural baseline models. We also provide the results obtained for the ASR outputs. Additionally, we will show how much improvement did the knowledge transfer method give. We used the micro F1 score evaluation metric [22] in all the experiments.

The final results for the Digitoday dataset are presented in Table 4 and for the out-of-domain Wikipedia test set in Table 5. In Table 6 we can see how well our model performs on the Parliament and Yle Pressiklubi datasets, annotated by the rule-based system.

Since the Parliament dataset is lowercased and without punctuation, during training, we simulated the same scenario and trained the model in that setting, which resulted in significant improvement. The results for the Parliament dataset when the training data is kept as it is (with capitalization and punctuation) is shown in Table 7.

To see how well our model agrees with the rule-based system, we evaluated the system only on entities that were found by that system. The results for the Parliament and Yle Pressiklubi datasets are shown in Table 8.

At the end, we manually annotated 50 sentences from both ASR datasets in order to see how well the system performs on gold standard data. The results from the manually annotated Parliament and Yle Pressiklubi datasets are presented in Table 9.

Table 7: Overall micro average precision, recall and F1 scores for the Parliament dataset, trained without removing capitalization and punctuation.

TAG	precision	recall	F1
PER	72.73	8.60	15.38
LOC	19.57	18.37	18.95
avg	29.82	11.97	17.09

Table 8: Overall micro average precision, recall and F1 scores for the Parliament and Yle Pressiklubi datasets, comparing only entities found by the rule-based system.

	Parliament data			Yle Pressiklubi data		
TAG	precision	recall	F1	precision	recall	F1
PER	98.81	89.25	93.79	85.04	85.71	85.38
LOC	100.00	69.39	81.93	89.55	86.96	88.24
ORG	/	/	/	78.95	26.79	40.00
avg	99.15	82.39	90.00	85.92	72.91	78.88

Table 9: Overall micro average precision, recall and F1 scores for the manually annotated Parliament and Yle Pressiklubi datasets.

	Parliament data			Yle Pressiklubi data		
TAG	precision	recall	F1	precision	recall	F1
PER	91.43	84.21	87.67	91.11	85.42	88.17
LOC	77.27	80.95	79.07	84.62	84.62	84.62
ORG	/	/	/	100.00	32.14	48.65
avg	85.96	83.05	84.48	90.00	70.59	79.12

6 ANALYSIS OF THE RESULTS

From Table 4 we can see that when we added transferred tags from Estonian language, we gained a slight boost in the F1 score. Our model achieved F1 score of 84.73, which is slightly better than the GÜNGÖR-NN architecture. Still, our system performed worse than the rule-based FiNER system, which achieved F1 score of 86.82.

In Table 5 we can see the results for the Wikipedia test set. On this out-of-domain dataset, the knowledge transfer technique improved the F1 score from 62.98 to 67.66. Compared to the GÜNGÖR-NN architecture our system did far better but it still falls behind compared to the FiNER system. From the results in Tables 4 and 5 we can see that transferring tags from Estonian had bigger impact on the out-of-domain Wikipedia set than on the Digitoday test set. We can also observe that neural network architectures suffer more from out-of-domain data but our architecture still performs better than the GÜNGÖR-NN.

If we compare the results presented in Table 6, we can see that our systems has low precision for the Parliament data when evaluated against the rule-based system annotations. The reason is that our system is able to find more entities than the rule-based system and since those entities are not present in the annotations obtained by that system, we get high number of false positives.

When comparing only with the entities found by the rule-based system, we can see that our system agrees with the rule-based system almost all the time, which results in high precision.

When evaluated on the manually annotated data, we can see that our system achieves relatively good results.

7 CONCLUSION

In this paper we showed that our system which incorporates word, character and morph representations achieves competitive results on Digitoday dataset. Furthermore, we saw that transferring tags from Estonian language using multilingual embeddings significantly improved the results on the out-of-domain Wikipedia test set.

Additionally, we evaluated our system on two ASR output datasets, where one of them did not have capitalization and punctuation, which caused difficulties for our system. In order to mitigate those difficulties, we converted our training set to lowercase and removed the punctuation in order to simulate ASR setting, which yielded significant improvement.

ACKNOWLEDGMENTS

This work was supported by the Kone Foundation. This work was supported by the Academy of Finland (grant 329267) and EU's Horizon 2020 research and innovation programme via the project MeMAD (GA 780069). The computational resources were provided by Aalto ScienceIT.

REFERENCES

[1] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research* 12, Aug (2011), 2493–2537.

[2] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing (TSLP)* 5, 1 (2007), 3.

[3] Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. 2000. Rule-based named entity recognition for Greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*. 75–78.

[4] Xiaocheng Feng, Xiachong Feng, Bing Qin, Zhangyin Feng, and Ting Liu. 2018. Improving Low Resource Named Entity Recognition using Cross-lingual Knowledge Transfer.. In *IJCAI*. 4071–4077.

[5] Onur Güngör, Suzan Üsküdarlı, and Tunga Güngör. 2018. Improving Named Entity Recognition by Jointly Learning to Disambiguate Morphological Tags. *arXiv preprint arXiv:1807.06683* (2018).

[6] Teemu Hirsimaki, Janne Pylkkonen, and Mikko Kurimo. 2009. Importance of high-order n-gram models in morph-based speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 4 (2009), 724–732.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[9] Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *proceedings of ACL-08: HLT*. 407–415.

[10] Onur Kuru, Ozan Arkan Can, and Deniz Yuret. 2016. Charner: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 911–921.

[11] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[12] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the Variance of the Adaptive Learning Rate and Beyond. *arXiv preprint arXiv:1908.03265* (2019).

[13] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354* (2016).

[14] Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 188–191.

[15] Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, 157–176.
[16] Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2019. A Finnish news corpus for named entity recognition. *Language Resources and Evaluation* (2019), 1–26.
[17] Cicero Nogueira dos Santos and Victor Guimaraes. 2015. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008* (2015).
[18] Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*. 107–110.
[19] Vesa Siivola, Teemu Hirsimaki, Mathias Creutz, and Mikko Kurimo. 2003. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Eighth European Conference on Speech Communication and Technology*.
[20] Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 21–24.
[21] Peter Smit, Sami Virpioja, Mikko Kurimo, et al. 2017. Improved Subword Modeling for WFST-Based Speech Recognition.. In *INTERSPEECH*. 2551–2555.
[22] Cornelis Joost Van Rijsbergen. 1979. Information retrieval. (1979).
[23] Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. *arXiv preprint arXiv:1808.09861* (2018).

[b]0.49

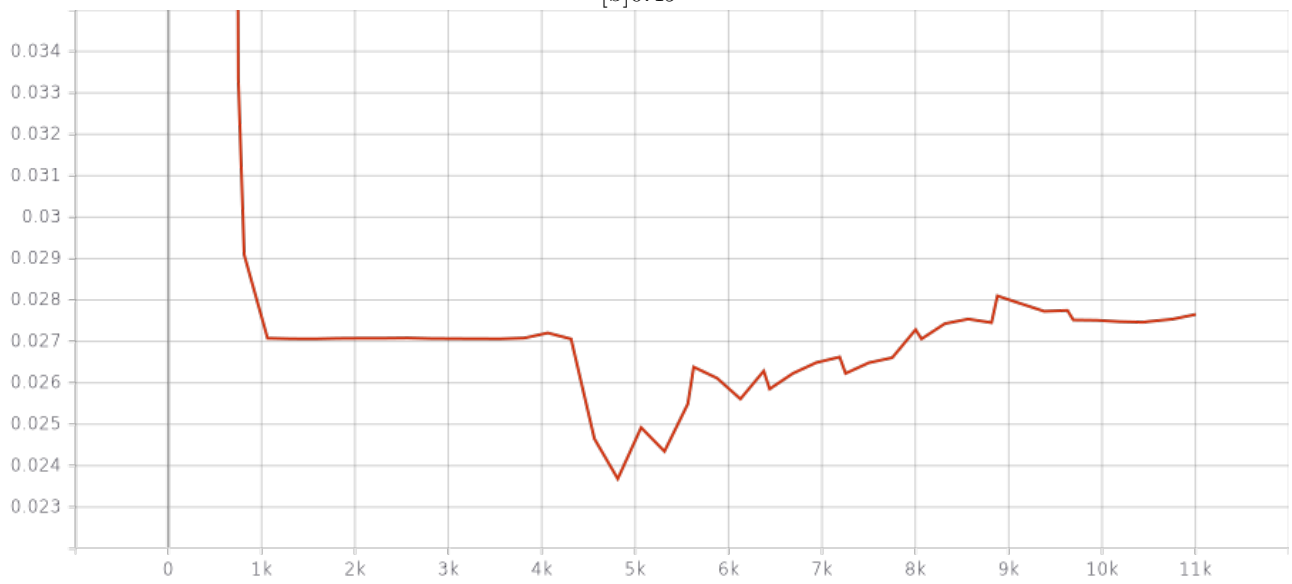

Figure 13: Base model validation loss

[b]0.49

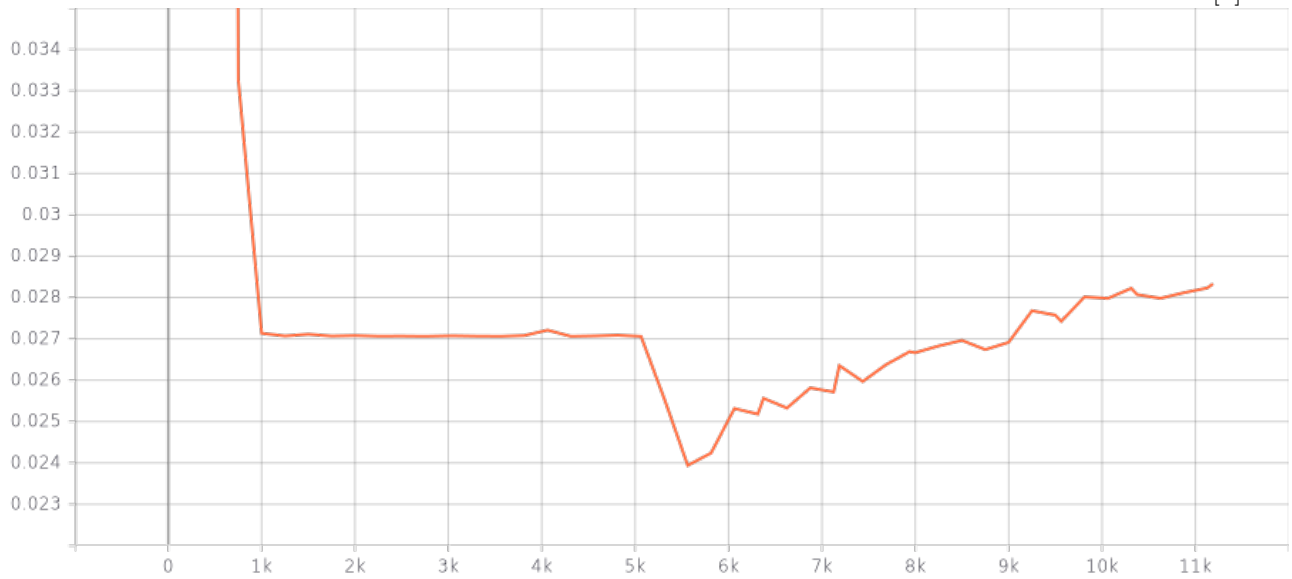

Figure 14: VQA model validation loss

[b]0.49

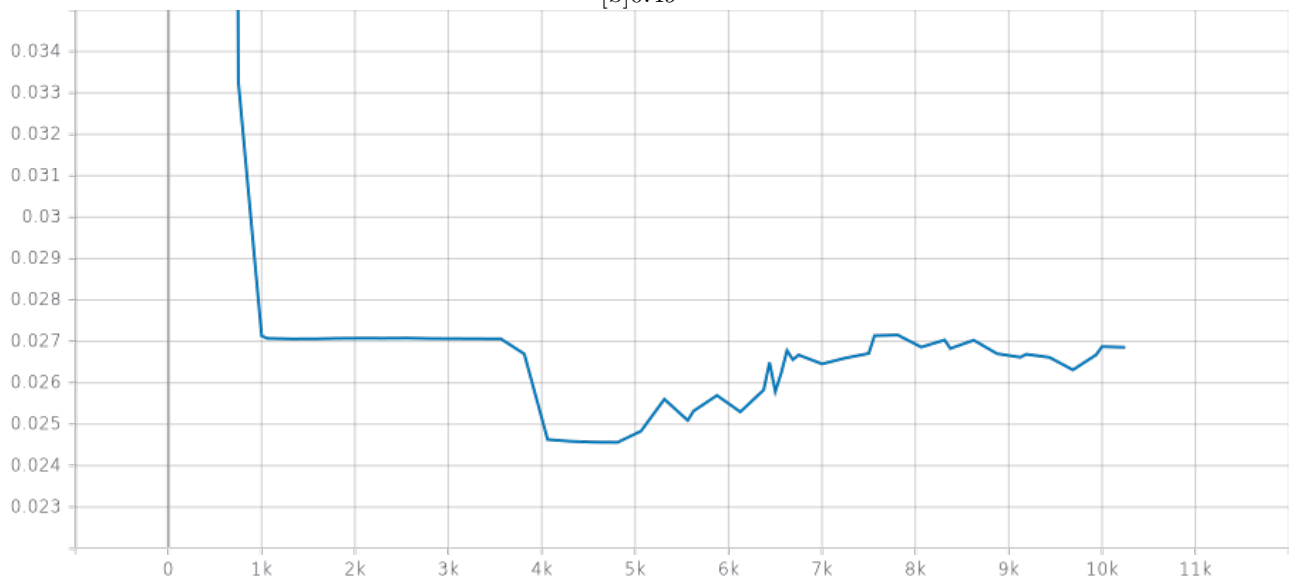

Figure 15: NLVR2 model validation loss

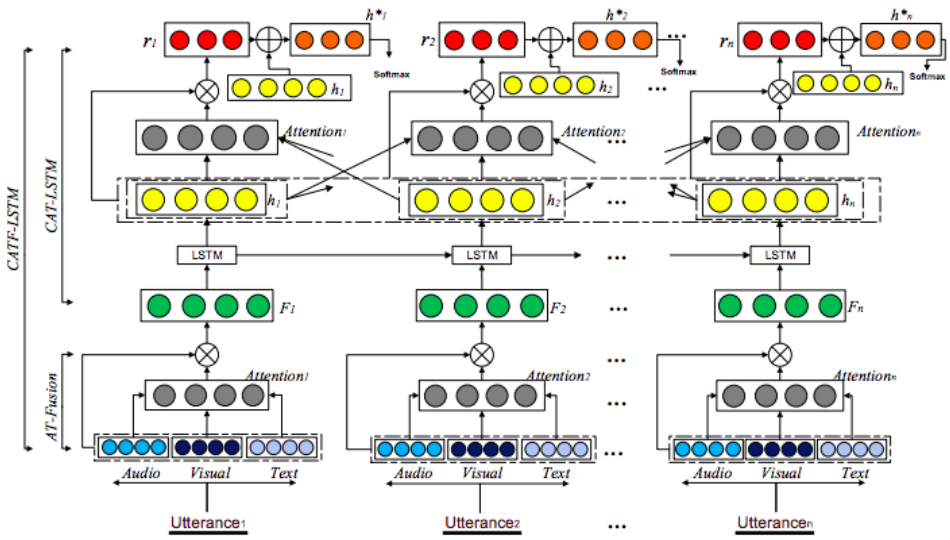

Figure 16: Taken from[48] CATF-LSTM takes input from multiple modalities, fuses them using AT-Fusion, and sends the output to CAT-LSTM for classification.

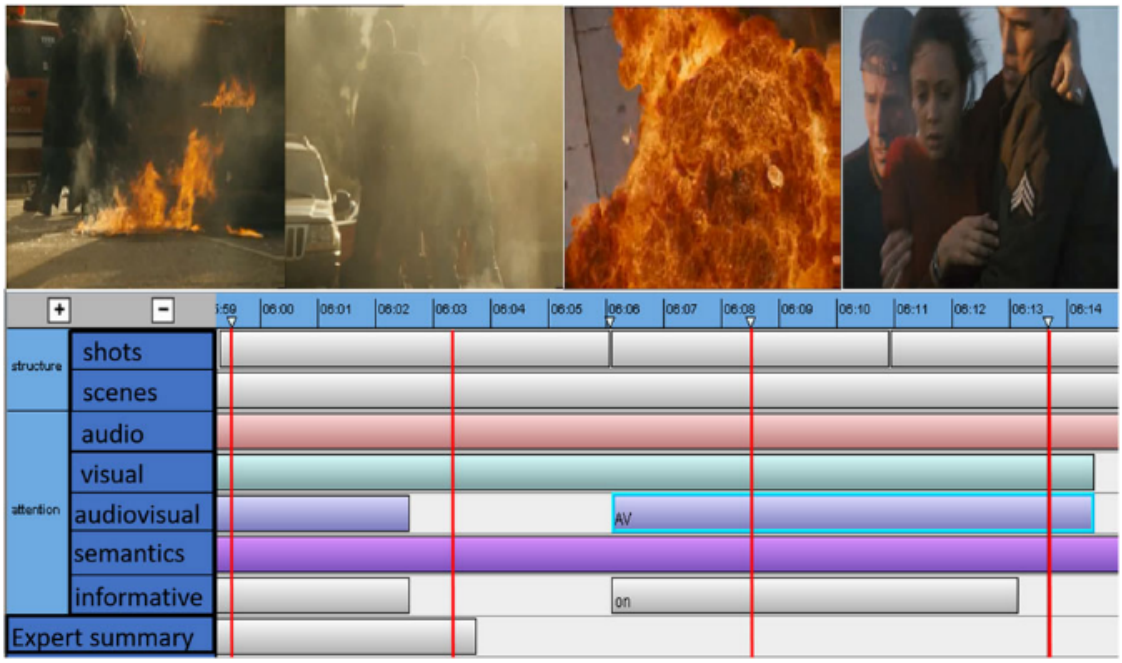

Figure 17: Interface for saliency annotation with its annotation layers. The red lines represent the time instances for each key frames which were categorized as either salient or not salient

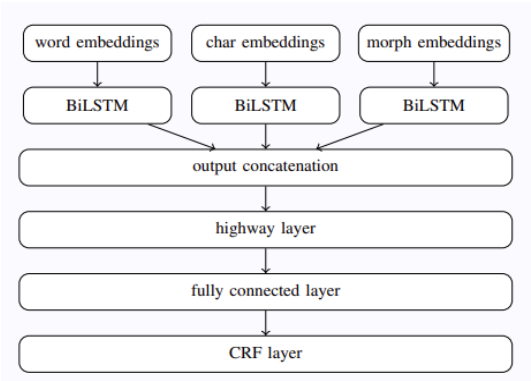

Figure 18: LSTM-CRF model that utilizes char, morph and word embeddings.

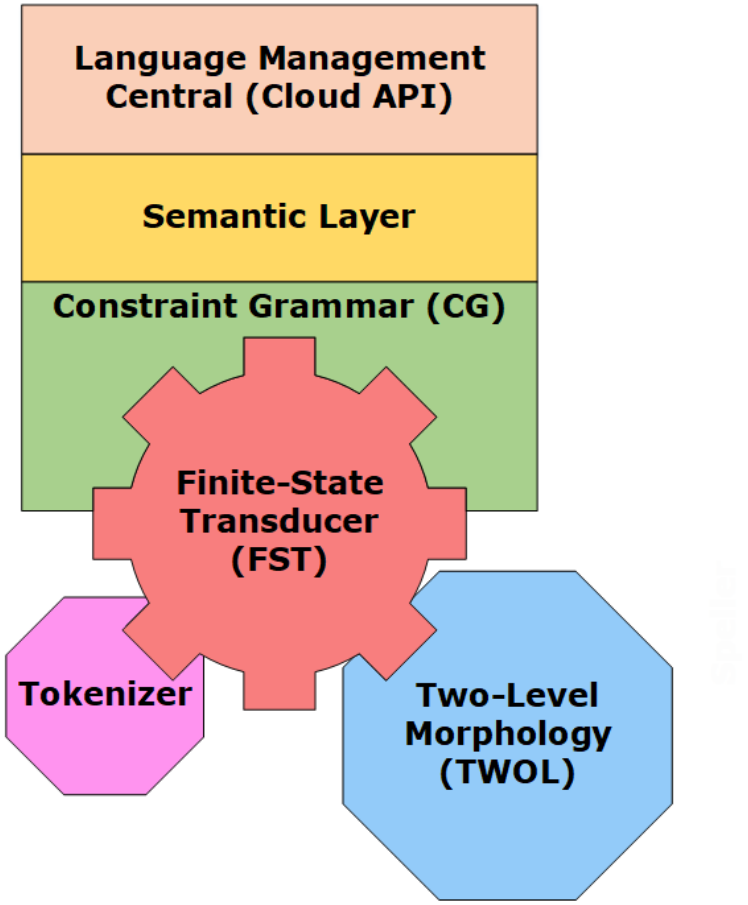

Figure 19: A schematic example of Lingsoft analyzer components.

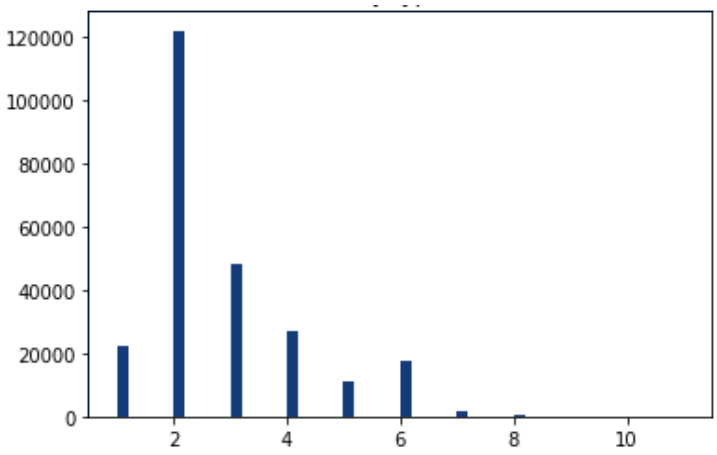

Figure 20: Distribution of the number of mention type attributed to each named entity

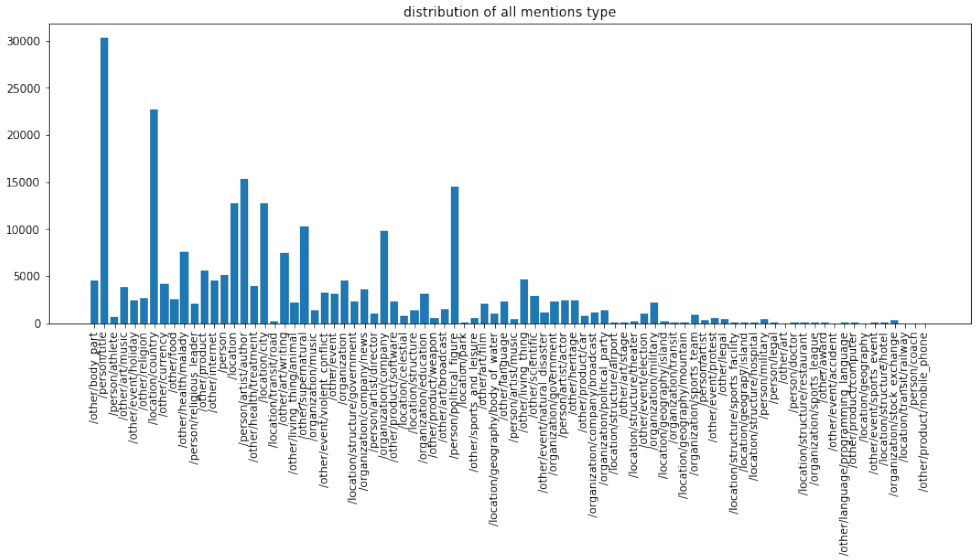

Figure 21: Distribution of the mention types

	start	end	sentence	mention_type
0	18	19	Usually directors , otherwise , they have bear...	[/other, /other/body_part]
1	1	2	Usually directors , otherwise , they have bear...	[/person/title, /person]
2	0	1	Oh .	[/person/athlete, /person]
3	0	1	Oh ! I had n't thought about that .	[/person/athlete, /person]
4	0	1	Actually a shaved head is really hard to manag...	[/other, /other/art, /other/art/music]
5	8	9	Actually a shaved head is really hard to manag...	[/person/title, /person]
6	3	4	Actually a shaved head is really hard to manag...	[/other, /other/body_part]
7	7	8	Do you have to shave it every day ?	[/other, /other/event, /other/event/holiday]

Figure 22: Example of the Ontonotes Data

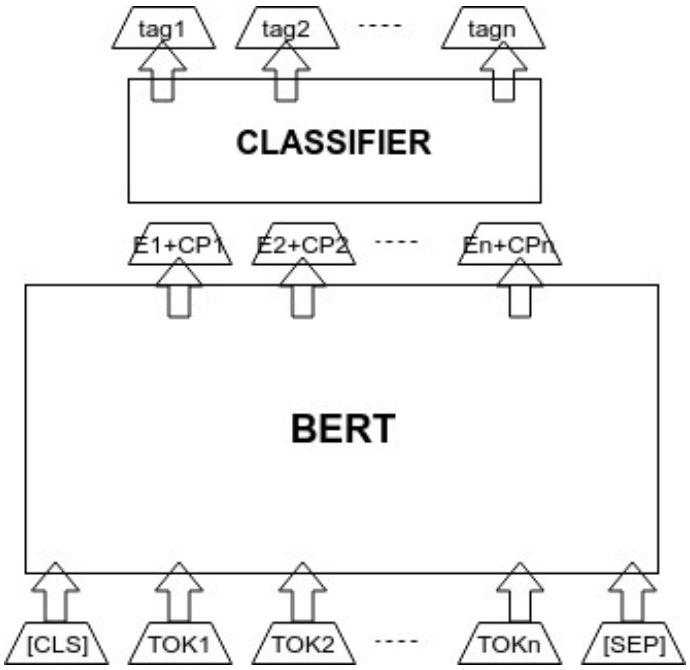

Figure 23: Model diagram

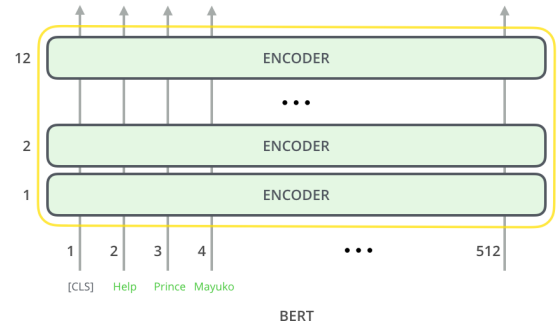

Figure 24: BERT's architecture

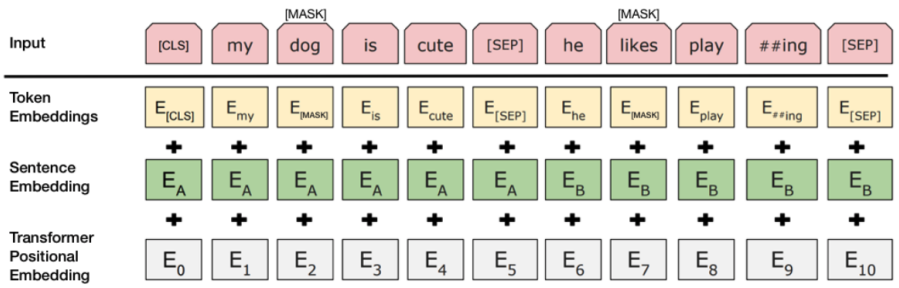

Figure 25: The input format that BERT expects

	precision	recall	f1-score	support
location/celestial	0.96	0.29	0.45	92
location/city	0.55	0.51	0.53	1311
location/country	0.61	0.81	0.69	2121
location/geography	1.00	0.19	0.32	172
location/geograpy	0.00	0.00	0.00	10
location/park	1.00	0.14	0.25	7
location/structure	0.54	0.30	0.39	581
location/transit	1.00	0.07	0.14	40
organization/company	0.57	0.87	0.69	1836
organization/education	0.97	0.39	0.55	379
organization/government	0.76	0.17	0.28	244
organization/military	0.79	0.18	0.29	271
organization/music	0.89	0.27	0.41	126
organization/political_party	0.96	0.19	0.31	145
organization/sports_league	0.00	0.00	0.00	19
organization/sports_team	0.86	0.13	0.23	91
organization/stock_exchange	0.88	0.33	0.48	85
organization/transit	0.00	0.00	0.00	15
other/art	0.54	0.52	0.53	1670
other/award	0.00	0.00	0.00	12
other/body_part	0.78	0.43	0.56	467
other/currency	0.52	0.30	0.38	431
other/event	0.35	0.54	0.42	1396
other/food	0.95	0.07	0.13	277
other/health	0.29	0.47	0.36	1152
other/heritage	0.63	0.18	0.28	246
other/internet	0.59	0.71	0.64	452
other/language	0.96	0.24	0.38	210
other/legal	0.00	0.00	0.00	70
other/living_thing	0.63	0.31	0.42	695
other/product	0.33	0.32	0.32	1056
other/religion	0.89	0.37	0.52	267
other/scientific	0.71	0.43	0.54	303
other/sports_and_leisure	1.00	0.39	0.56	57
other/supernatural	0.93	0.77	0.84	1078
person/artist	0.56	0.58	0.57	2202
person/athlete	1.00	0.15	0.26	82
person/doctor	0.00	0.00	0.00	11
person/legal	0.00	0.00	0.00	8
person/military	1.00	0.11	0.21	61
person/political_figure	0.74	0.61	0.67	1525
person/religious_leader	0.91	0.51	0.65	229
person/title	0.66	0.82	0.73	3309
accuracy			0.57	24811
macro avg	0.64	0.32	0.37	24811
weighted avg	0.61	0.57	0.56	24811

Figure 26: Results breakdown for the SVM classifier on Ontonotes

Layer (type)	Output Shape	Param #
flatten_11 (Flatten)	(None, 776)	0
dense_21 (Dense)	(None, 256)	198912
dropout_5 (Dropout)	(None, 256)	0
dense_22 (Dense)	(None, 44)	11308

Total params: 210,220
Trainable params: 210,220
Non-trainable params: 0

Figure 27: DNN architecture

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 776, 1)]	0
conv1d_6 (Conv1D)	(None, 774, 128)	512
max_pooling1d_6 (MaxPooling1	(None, 387, 128)	0
dropout_8 (Dropout)	(None, 387, 128)	0
conv1d_7 (Conv1D)	(None, 385, 64)	24640
max_pooling1d_7 (MaxPooling1	(None, 192, 64)	0
dropout_9 (Dropout)	(None, 192, 64)	0
conv1d_8 (Conv1D)	(None, 190, 32)	6176
max_pooling1d_8 (MaxPooling1	(None, 95, 32)	0
dropout_10 (Dropout)	(None, 95, 32)	0
flatten_2 (Flatten)	(None, 3040)	0
dense_4 (Dense)	(None, 256)	778496
dropout_11 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 44)	11308

Total params: 821,132
Trainable params: 821,132
Non-trainable params: 0

Figure 28: CNN architecture

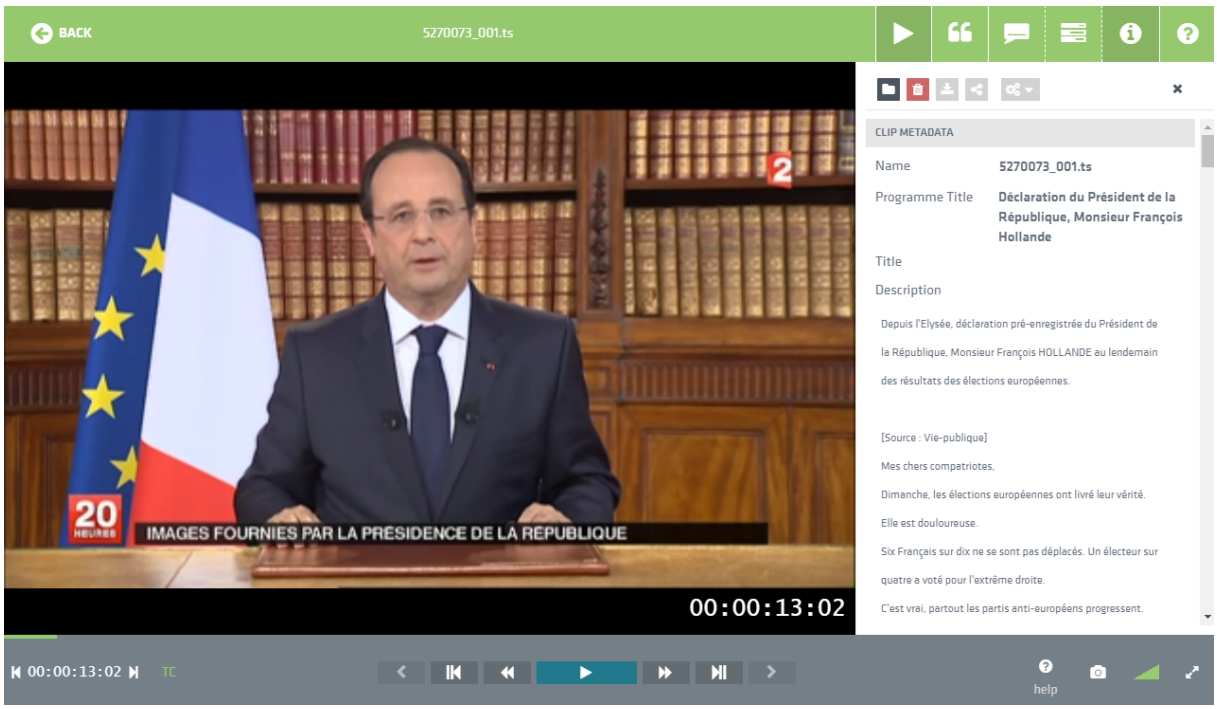

Figure 29: A screenshot for the program in the Flow platform

Text

donc voici cette intervention mes chers compatriotes dimanche les élections européennes ont livré leur vérité elle est douloureuse six français sur dix ne sont pas déplacé un électeur sur quatre à voter pour l'extrême droite c'est vrai par tous les partis européens progresse mais c'est en france pays fondateurs de l'union européenne patrie des droits de l'homme au pays des libertés que l'extrême droite arrive aussi largement en tête bien sûr ce vote n'efface pas tous les suffrages ceux qui se sont portés notamment sur les partis européens mais ce vote il est là et il doit être regardés en face c'est ce que je fais comment l'interpréter ce vote c'est une défiance à l'égard de l'europe qui inquiète plus qu'elle ne protège c'est une défiance à l'égard des partis de gouvernement de la majorité comme de l'opposition ce vote c'est une défiance à l'égard de la politique qui après tant d'années de crise appellent toujours des fort ce qu'on le revoit encore les résultats ce serait une faute et je ne commettrai pas que de fermer les yeux sur cette réalité parce qu'elle traduit une peur du déclin de la france de la mondialisation et ce sentiment exprimé tant de fois d'abandon face à la dureté de la vie mais le pire le pire ce serait de renoncer à ce qui fait la france ces valeurs sont grands sous une influence son ambition sa place en europe et dans le monde nous sommes un grand pays et ne peut concevoir son destin dans le repli dans la fermeture dans le rejet l'europe elle ne peut pas avancer sur la france mais l'avenir de la france il est en europe je suis européen mon devoir c'est de réformer la france et de réorienter l'europe l'europe elle a réussi notamment depuis deux ans à surmonter la crise de la zone euro elle était proche de l'éclatement mais à quel prix celui d'une austérité qui a fini par décourager les peuples aussi demain pas plus tard que demain au conseil européen gérard affirmerait que la priorité c'est la croissance c'est l'emploi c'est l'investissement l'europe elle est devenue illisible j'en suis conscient lointaine et pour tout dire incompréhensible même pour les états ça ne peut plus durer l'europe elle doit être simple claire pour être efficace là où elle est attendue et se retirer là où elle n'est pas nécessaire l'europe elle doit préparer l'avenir les nouvelles technologies la transition énergétique et sa propre défense elle doit protéger ses frontières ses intérêts ses valeurs sa culture tel doit être le mandat qui sera confiée à la prochaine commission européenne et j'y veillerai mais pour parler d'une voix forte la france doit elle même être forte depuis dix ans elle perd ses emplois et notamment dans l'industrie sa compétitivité se dégrade son déficit commercial se creusent depuis dix ans la france à cause de politiques qui n'ont pas été conduit à la accumulé des dettes c'est pas l'europe qui nous demande de faire des réformes c'est pour la france que nous devons les mener à bien et c'est ce que j'ai décidé en confiant au gouvernement de manuel valls sa feuille de route quelle est elle c'est l'emploi par le soutien aux entreprises le pacte de responsabilité c'est le pouvoir d'achat par des baisses d'impôts c'est la justice sociale par l'arrêt de priorité répété réaffirmé à l'éducation c'est la simplification la modernisation et ce sera tout l'enjeu de la réforme de notre organisation territoriale de grandes régions avec une évolution de nos collectivités et ce sera présenté dès la semaine prochaine cette ligne de conduite elle ne peut pas dévier en fonction des circonstances il faut de la constance de la ténacité du courage mais aussi de la rapidité dans la mise en oeuvre parce que les français ne peuvent pas attendre l'avenir de nos institutions elles sont solides elle nous donne les moyens d'agir et au bout du chemin j'en suis convaincu mais il faudra le démontrer ce sera la réussite et la réussite de tous dans les épreuves face au défi le rassemblement il est nécessaire le rassemblement des français ce qui nous unit c'est notre attachement à la démocratie à la république ce qui nous unit au-delà de tout c'est notre amour de la france et ce sera le combat que je mènerai tout au long de mon quinquennat vive la république et vive la france

Annotations

PR	Annotation	Annotation (en)	
0.0129	Zone euro	Eurozone	>>
0.0100	Crise de la dette dans la zone euro	European debt crisis	>>
0.0090	Commission européenne	European Commission	>>
0.0081	Mondialisation	Globalization	>>
0.0071	Transition énergétique	Energy transition	>>
0.0065	Justice sociale	Social justice	>>
0.0064	Droits de l'homme	Human rights	>>
0.0061	Balance commerciale	Balance of trade	>>
0.0059	Démocratie	Democracy	>>
0.0059	Parti politique européen	European political party	>>
0.0055	Pouvoir d'achat	Purchasing power	>>
0.0052	Europe	Europe	>>
0.0052	Quinquennat (politique)	Quinquennat	>>
0.0047	Conseil européen	European Council	>>
0.0044	État	State (polity)	>>
0.0042	Politique	Politics	>>
0.0041	Culture	Culture	>>
0.0041	Politique de rigueur	Austerity	>>
0.0039	République	Republic	>>
0.0039	Croissance économique	Economic growth	>>

Figure 30: The results of running Wikifier on the ASR transcript of the speech

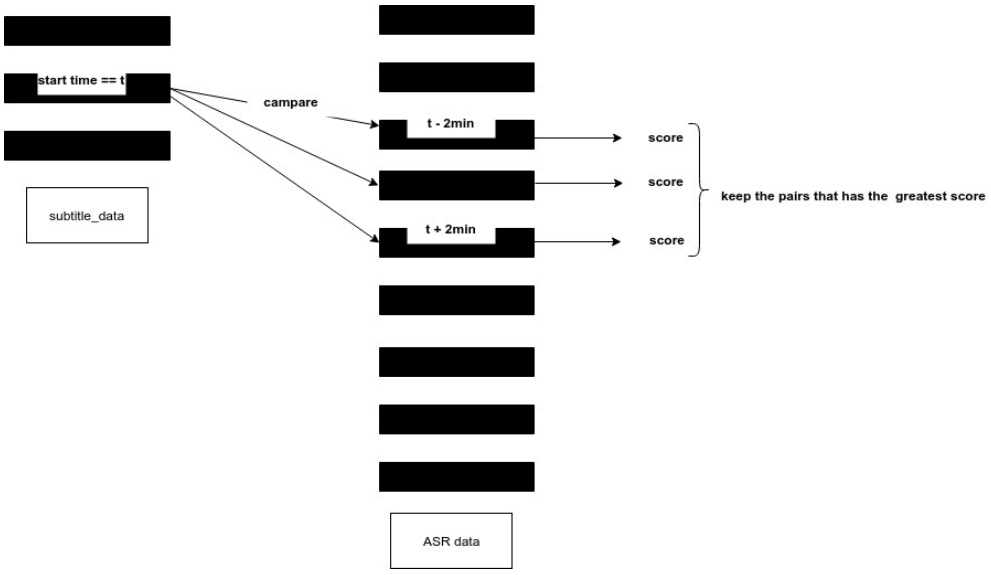

Figure 31: the strategy for aligning ASR with manual subtitle

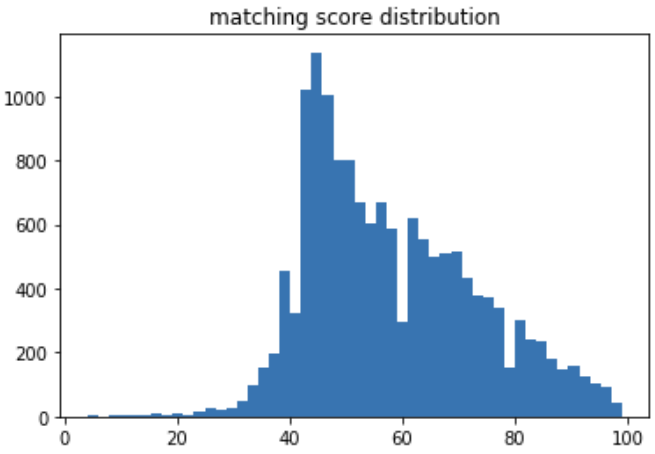

Figure 32: Matching scores distribution