Automatic Multilingual Multimedia Summarization and Information Retrieval

Carlos Gonzalez-Gallardo

ACADÉMIE D'AIX-MARSEILLE
AVIGNON UNIVERSITÉ

THESIS

presented at the Avignon Université to obtain
the degree of Doctor

SPECIALITY : Computer Science

École Doctorale 536 « Agrosciences et Sciences »
Laboratoire Informatique d'Avignon (EA 4128)

Automatic Multilingual Multimedia Summarization and Information Retrieval

Résumé automatique multimédia et multilingue et Recherche d'information

by
Carlos-Emiliano GONZÁLEZ-GALLARDO

Defended publicly on December 18, 2019 before the jury members:

M^{me}	Violaine PRINCE	Professor, LIRMM, Montpellier 2	Rapporteur
M.	Eric GAUSSIER	Professor, LIG, Grenoble	Rapporteur
M^{me}	Fatiha SADAT	Professor, GDAC, Montréal	Examiner
M.	Laurent BESACIER	Professor, LIG, Grenoble	Examiner
M.	Kamel SMAILI	Professor, LORIA, CNRS-Lorraine-Inria	Examiner
M.	Juan-Manuel TORRES-MORENO	Associate Professor HDR, LIA, Avignon	Advisor
M.	Eric SANJUAN	Associate Professor HDR, LIA, Avignon	Co-Advisor

Laboratoire Informatique d'Avignon

Abstract

As multimedia sources have become massively available online, helping users to understand the large amount of information they generate has become a major issue. One way to approach this is by summarizing multimedia content, thus generating abridged and informative versions of the original sources. This PhD thesis addresses the subject of text and audio-based multimedia summarization in a multilingual context. It has been conducted within the framework of the Access Multilingual Information opinionS (AMIS) CHISTERA-ANR project, whose main objective is to make information easy to understand for everybody.

Text-based multimedia summarization uses transcripts to produce summaries that may be presented either as text or in their original format. The transcription of multimedia sources can be done manually or automatically by an Automatic Speech Recognition (ASR) system. The transcripts produced using either method differ from well-formed written language given their source is mostly spoken language. In addition, ASR transcripts lack syntactic information. For example, capital letters and punctuation marks are unavailable, which means sentences are nonexistent. To deal with this problem, we propose a Sentence Boundary Detection (SBD) method for ASR transcripts which uses textual features to separate the Semantic Units (SUs) within an automatic transcript in a multilingual context. Our approach, based on subword-level information vectors and Convolutional Neural Networks (CNNs), overperforms baselines by correctly identifying SU borders for French, English and Modern Standard Arabic (MSA). We then study the impact of cross-domain datasets over MSA, showing that tuning a model that was originally trained with a big out-of-domain dataset with a small in-domain dataset normally improves SBD performance. Finally, we extend ARTEX, a state-of-the-art extractive text summarization method, to process documents in MSA by adapting preprocessing modules. The resulting summaries can be presented as plain text or in their original multimedia format by aligning the selected SUs.

Concerning audio-based summarization, we introduce an extractive method which represents the informativeness of the source based on its audio features to select the segments that are most pertinent to the summary. During the training phase, our method uses available transcripts of the audio documents to create an informativeness model which maps a set of audio features with a divergence value. Subsequently, when summarizing new audio documents, transcripts are not needed anymore. Results over a multi-evaluator scheme show that our approach provides understandable and infor-

mative summaries.

Evaluation measures is also a field which we deal with. We develop Window-based Sentence Boundary Evaluation (WiSeBE), a semi-supervised metric based on multi-reference (dis)agreement, that questions if evaluating an automatic SBD system based on a single reference is enough to conclude how well the system is performing. We also explore the possibility of measuring the quality of an automatic transcript based on its informativeness. In addition, we study to what extent automatic summarization may compensate for the problems raised during the transcription phase. Lastly, we study how text informativeness evaluation measures may be extended to passage interestingness evaluation.

Contents

Chapter 1

Introduction

Contents

Information age, marked by the digital revolution of the mid 20th century, has changed the way we share and consume information. However, massive dissemination of information has its basis on three transcendent events (Figure 1.1):

1. The invention of the modern printing press by Johannes Gutenberg (1400-1468) in the 15th century, enabling the fast dissemination of information throughout Europe (Palermo, Elizabeth, 2014).

2. The first radio broadcast for entertainment and music on 24 December 1906 by Reginald Aubrey Fessenden (1866-1932) (Sterling, Christopher H. and Skretvedt, Randy , 2019).

3. The first public demonstration of television by the Scottish engineer John Logie Baird (1888-1946) in 1931 (Cité de l'Économie, 2019).

From the invention of modern printing press to television shows, information has been a centralized resource which depends of principal actors to be massively shared. Also, interaction between different media formats is limited to the form it is disseminated. Nowadays, with the Internet and the World Wide Web created in 1989, a new way of creating, sharing and consuming information is available. Information can be shared massively and fast by independent entities while multimedia content is the standard. Multimedia refers to the incorporation of different content forms such as text, audio and video, which provide a more complete and informative experience.

Figure 1.1: *Information Dissemination Actors: (A) Johannes Gutenberg, (B) Reginald Aubrery Fessenden and (C) John Logie Baird*

To illustrate how the Internet and multimedia content are shaping the way information is created, shared and consumed we recall the fire occurred in the Notre-Dame's Cathedral rooftop on April 15, 2019[1]. At 18h43 the second smoke detector went off an the fire showed in the Notre-Dame's Cathedral rooftop. 16 minutes later, at 18h59 photos, videos and comments from the fire were circulating on social media platforms like Twitter[2,3]. At 19h014, half an hour later, newspapers websites worldwide where publishing notes about the event[4].

As shown in this example (Figure 1.2), only 31 minutes were necessary to make an event globally available in a sort of different formats. However, since multimedia sources have become massive thanks to online availability, the need of helping the understanding of the information they produce has become a major issue. One way to approach this demand is by creating an automatic summary of the multimedia information and then present the summarized content to the user in a format which may facilitate the understanding. But... what does an automatic summary is?

1.1 Automatic Multimedia Summarization

Some definitions to *summary* that can be found on various dictionaries in different languages are:

[1] https://www.bbc.com/news/world-europe-47941794

[2] `https://twitter.com/ecoursin/status/1117835281065967616`

[3] `https://twitter.com/carlosferiab/status/1117834790387036160`

[4] `http://www.leparisien.fr/paris-75/incendie-en-cours-a-la-\`
`cathedrale-notre-dame-de-paris-15-04-2019-8053935.php`

Figure 1.2: *Multimedia Information of the Notre-Dame's Cathedral Roof Fire*

According to the Oxford English Dictionary[5]:

"A brief statement or account of the main points of something"

According to the Larousse French Dictionary[6]:

"Abbreviated form of the contents of a text, a document or a film"[7]

According to the Spanish from Mexico Dictionary (DEM)[8]:

"Short spoken or written exposition from the most important ideas, aspects or parts of something"[9]

From the scientific point of view, Rush et al. (1971) defined a summary as a set of sentences produced by rejecting sentences of the original which are irrelevant to the abstract. Saggion and Lapalme (2002) define a summary as the condensed version of a source document which has a recognizable genre and the purpose of giving the reader an exact and concise idea of the source. For Sakai and Sparck-Jones (2001) a summary is a *"reductive transformation of a source text into a summary text by extraction or generation"*. Radev et al. (2002) establish that a summary is the result of creating a compressed version of a document with the most useful information for the user.

[5] `https://www.lexico.com/`
[6] `https://www.larousse.fr/dictionnaires/francais-monolingue/`
[7] *"Forme abrégée du contenu d'un texte, d'un document, d'un film"*
[8] `https://dem.colmex.mx/`
[9] *"Exposición breve, oral o escrita, de las ideas, aspectos o partes más importantes de algo"*

Finally, Saggion and Poibeau (2013) approach summarization from a text perspective. They refer to text summarization as the reduction of a text to its essential content.

By general rule, a summary is supposed to be shorter that its source document(s). The size of a summary is defined by a Compression Ratio (CR) $\tau \in [0, 1]$ multiplied by the length of its source document (Equation 1.1), expressed in characters, words or sentences.

$$|summary| = \tau \times |source| \tag{1.1}$$

For a document containing 200 sentences, a $\tau = 0.25$ will produce a 50 sentences summary. CRs between 0.15 and 0.30 show to improve performance of automatic summarization systems (Lin, 1999).

Automatic summarization refers to the fact that the summarization process is produced by a computer program. Torres-Moreno (2014) defines an automatic summary as a text generated by a software that is coherent, contain a significant amount of relevant information from the source text and its CR is less than a third of the length of the source document.

From dictionary to scientific definitions, we identify six elements a summary contemplate.

- **Source:** The original document(s) to summarize; it can refer to a text, audio, video, film, etc.
- **Mechanism:** The way the summary is created; manually by an expert or automatically by a computer software.
- **Method:** The approach that is followed to create the summary. The two main types of methods are extractive and abstractive. In the extractive approach, a set of segments of the source document(s) is selected to conform the summary. By contrast, the abstractive approach produces summaries in which some of its material is not present in the source document (Mani and Maybury, 2001).
- **Compression rate:** The ratio between the length of the summary and the length of the source document.
- **Intention:** The purpose of the summary; to alert or inform the user providing an exact and concise idea of the source.
- **Output:** The form the summary is presented. A summary of a video can be presented in multiple forms depending of the need: video, audio, text, video+text, etc.

In a multimedia context, the summarization process is conducted different depending of the source type and the desired output format.

- **Text-based Multimedia Summarization**: Summarization is performed with text summarization techniques without extra audio or video information. If the source

document is an audio or a video (with audio), a manual or automatic transcription process is performed to obtain the corresponding text representation. If desired and if time information is available, the resulting text summary can be aligned to its original source to present the summary on its original format. This type of summarization process is very informative and useful in cases where it is necessary to focus on the things that are said. Nevertheless, it is essential to count with a textual representation of the source, which depending on the situation it may be difficult to obtain.

- **Audio-based Multimedia Summarization**: Summarization is performed with audio summarization techniques without extra text or video information. A posterior transcription process may be performed to complement the resulting audio summary. This type of summarization process is convenient when it is needed to focus on how things are said or to highlight salient moments within the audio source. If the source has no salient audio, this type of summarization method may not be very useful.
- **Video-based Multimedia Summarization**: Summarization is performed exclusively with video summarization techniques without extra text or audio information. Summaries result of this summarization process are visually informative and agreeable. However, audio discontinuities are likely to happen leading to understanding issues.

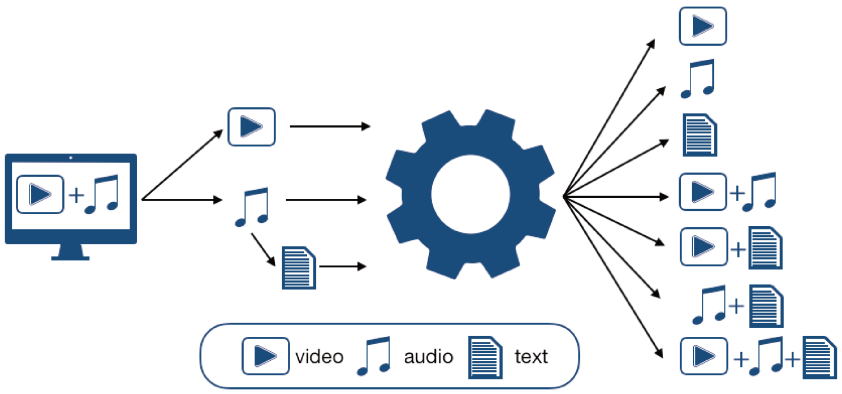

Figure 1.3: *Multimedia Summarization*

Figure 1.3 shows a general flow of the automatic multimedia summarization process. The source corresponds to a video+audio document; then, a transcript is produced from the audio component by an Automatic Speech Recognition (ASR) system. Depending from the summarization process (text, audio or video based), the resulting summary can be presented in the following formats: video, audio, text, video+audio, video+text, audio+text and video+audio+text.

This PhD thesis addresses the subject of text and audio-based multimedia summarization in a multilingual context. It has been developed in the frame of the Access Multilingual Information opinionS (AMIS) CHISTERA-ANR project, which aims to make information easy to understand to everybody.

1.2 Access Multilingual Information opinionS (AMIS)

With the growth of the information in different media such as TV programs or Internet, a new issue arises. How a user can access to the information which is expressed in a foreign language? This is the central question the Access Multilingual Information opinionS (AMIS)[10] project tries to answer.

AMIS is a CHIST-ERA[11] founded international project conceived in the frame of the Human Language Understanding: Grounding Language Learning (HLU) 2014's call. Its main objective is to make available a system which helps people to understand the content of a source video in a language the user does not understand by presenting its main ideas in a target understandable language (Smaïli et al., 2018). AMIS considers French and Modern Standard Arabic (MSA) as sources languages, while English the target language.

The four different architectures proposed in AMIS to summarize a video in a target language are shown in Figure 1.4. Each architecture follows a different approach and has advantages and disadvantages over the others. The purpose (**intention**) of all architectures is to inform the user by creating extractive (**method**) automatic summaries (**mechanism**) from videos (**source**) and presenting them in video format (**output**) in the target language.

- **SC1**: The summarization process is done directly over the video (visual features) with no audio and text content information. Then, an Automatic Speech Recognition (ASR) process is performed over the summary and the resulting transcript is translated to the target language and integrated as subtitles in the resulting video.
- **SC2**: The summarization process is done over the audio signal (audio features) with no video and text content information. Then, an ASR process is performed over the summary and the resulting transcript is translated to the target language and integrated as subtitles in the resulting video.
- **SC3**: An ASR process is performed over the audio of the input video. Later, the resulting transcript is translated to the target language to be summarized (textual features) without any audio and video information. Finally, the resulting segments are extracted from the source video and the text is integrated as subtitles.
- **SC4**: An ASR process is performed over the audio of the input video. The resulting transcript is then summarized (text features) without any audio or video information and the resulting summary is translated to the target language. Finally, the resulting segments are extracted from the source video and the text is integrated as subtitles.

Four partners of three different countries constitute the AMIS consortium. Each partner develops certain modules of the architectures explained in Figure 1.4 to in-

[10] http://deustotechlife.deusto.es/amis/

[11] CHIST-ERA is a program for European Coordinated Research on Long-term Information and Communication Technologies (ICT) and ICT-based scientific challenges.

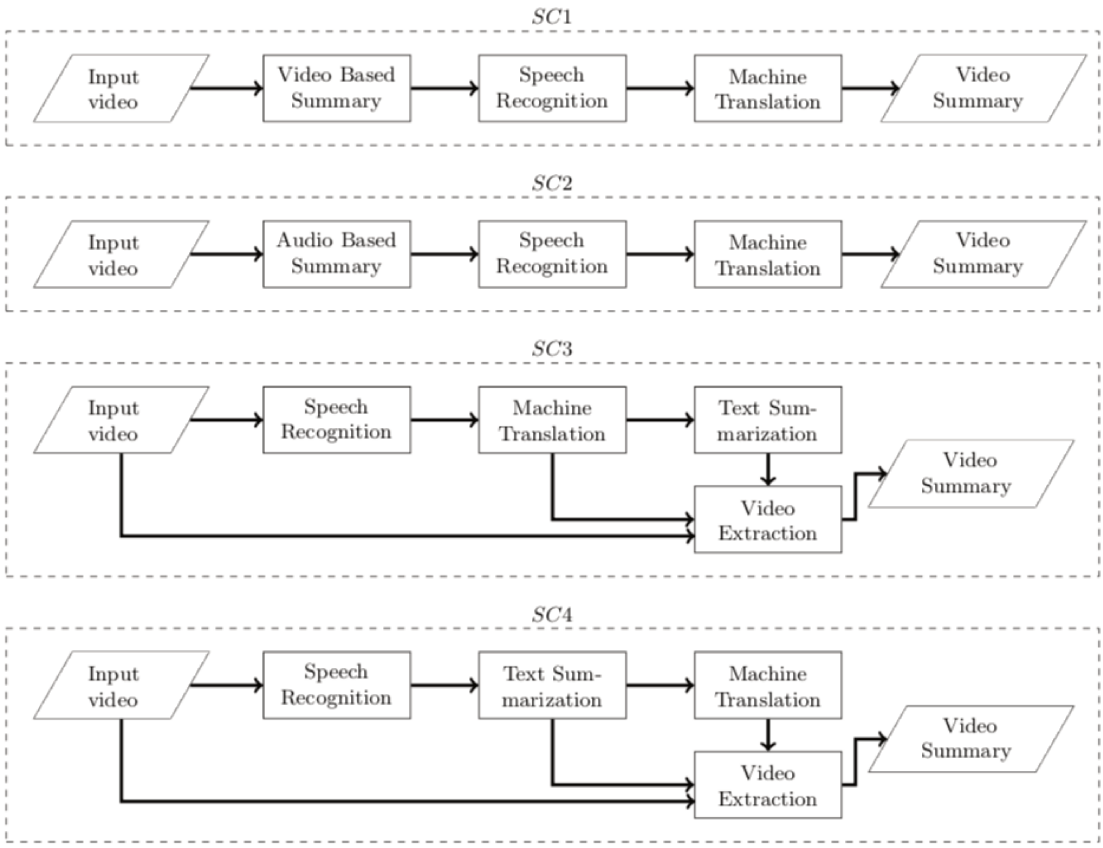

Figure 1.4: *AMIS Architectures for Summarizing a Video in a Target Understandable Language (Smaïli et al., 2018)*

teract in a collaborative strategy. Table 1.1 shows each partner, its country and the main components it develops. The research carried out in this thesis is closely related to the audio and text-based summarization components the Laboratoire Informatique d'Avignon (LIA) is in charge of. In addition, further work regarding Sentence Boundary Detection (SBD) and evaluation metrics has been done.

1.2.1 AMIS-Dataset

The *AMIS-Dataset* (Leszczuk et al., 2017) is a large corpus of newscasts and reports YouTube videos collected during the AMIS project. The AMIS-Dataset contains a total of 5 423 videos (310 hours) from 19 channels in French, English and Arabic. The length of the videos within the dataset varies between 1 and 64 minutes. Leszczuk et al. (2017) adopted the following process to create the *AMIS-Dataset*:

1. A list of controversial Twitter hashtags was identified
2. All tweets with the selected hashtags were extracted during a period of time

Institution	Country	Components
Université de Lorraine - LORIA	France	Speech Recognition and Machine Translation
Avignon Université - LIA	France	Audio-based Summarization and Text-based Summarization
AGH	Poland	Video-based Summarization
Universidad de Deusto	Spain	Evaluation

Table 1.1: *AMIS Partners and Components Distribution*

Hashtag	Arabic		English		French		Total	
	Videos	Time	Videos	Time	Videos	Time	Videos	Time
#animalrights	17	0.52	162	5.36	193	8.66	372	14.54
#deathpenalty	113	5.89	110	3.43	381	15.69	604	25.01
#extremeright	43	2	156	5.65	172	6.28	371	13.93
#fcbarcelona	91	6.56	13	0.42	13	0.31	117	7.29
#homosexualmarriage	81	6.83	97	5.09	37	2.47	215	14.39
#occupiedterritories	165	18.11	49	1.28	132	5.05	346	24.44
#realmadrid	357	16.3	24	1.43	32	2.18	413	19.91
#syria	419	38.59	237	13.78	542	31.08	1 198	83.45
#trump	178	11.04	874	48.73	463	24.14	1 515	83.91
#womenright	39	3.95	152	15.53	81	4.41	272	23.89
Total	1 503	109.79	1 874	100.71	2 046	100.26	5 423	310.76

Table 1.2: *AMIS-Dataset Breakdown*

3. Extracted tweets were filtered to keep only those tweets that contained valid YouTube links
4. Videos corresponding to YouTube links were downloaded and stored

Table 1.2 presents the selected hashtags, number of videos per hashtag and the time in hours of the downloaded videos from the *AMIS-Dataset*.

1.3 Objectives

This PhD thesis has two general objectives. The first one is to provide a text and audio multimedia summarization framework with a multilingual perspective. To accomplish this, we have defined the following specific objectives:

- Provide a Sentence Boundary Detection (SBD) package to divide an Automatic Speech Recognition (ASR) transcript into Semantic Units (SUs) for English, French and Modern Standard Arabic (MSA).
- Provide a text summarizer in order to process MSA.

- Provide a method to create audio based extractive summaries for English, French and MSA.

The second objective is to open the discussion about evaluation metrics and the subjectivity of gold standards in certain Natural Language Processing (NLP) tasks. To accomplish this, we have define the following specific objectives:

- Provide a SBD evaluation package which does not relies on a unique gold standard but takes into account agreement and disagreement between multiple references.
- Propose an original evaluation protocol for ASR system performance which takes into account transcript informativeness and at the same time reduces the information loss ASR errors produce.
- Provide a novel approach of passage interestingness evaluation based on discrete and continuous text informativeness measures.

1.4 Contributions

Contributions of this thesis are grouped in two categories. The first one involves text and audio based summaries creation while the second one comprises evaluation metrics on Sentence Boundary Detection (SBD), Automatic Speech Recognition (ASR) transcripts and passages interestingness.

- Text and audio-based summaries creation:
  - We propose a SBD method to separate a transcript from an ASR system into Semantic Units (SUs) for French, English and Modern Standard Arabic (MSA). The method, based on textual features, subword-level information vectors and Convolutional Neural Networks (CNNs), predicts if a target word within a context window corresponds or not to the border between two SUs.
  - We develop ARTEX for Modern Standard Arabic (ARTEX-MSA), an Automatic Text Summarization (ATS) system to summarize documents in MSA. ARTEX-MSA is an extension of Autre Résumeur de TEXtes (ARTEX) (Torres-Moreno, 2012a), a state-of-the-art extractive text summarization method originally developed to create summaries in English, Spanish and French. To rank the pertinence of all sentence within the text, ARTEX-MSA computes an inner product between the lexical vector of each sentence (vocabulary used by a sentence) and the document vector (text topic). The summary is then generated by assembling the highest ranked sentences.
  - We introduce an audio-based extractive summarization method which represents the informativeness of the source document in terms of its audio features. The method selects those segments that are more pertinent to the summary.

- Evaluation metrics:
  - The first contribution pose the debate if evaluating an automatic SBD system against a unique reference is enough to conclude how well the system is performing. For this we propose Window-based Sentence Boundary Evaluation (WiSeBE), a semi-automatic multi-reference sentence boundary evaluation protocol based on the necessity of having a more reliable way for evaluating the SBD task. WiSeBE not only evaluates the performance of a system against all references, but also takes into account the agreement between them.
  - We propose a methodology to measure the quality of an automatic transcript in terms of its informativeness and in which grade automatic summarization may compensate the information loss raised during the transcription phase. This approach uses an ATS evaluation protocol without reference (based on the informative content), which computes the divergence between probability distributions of different textual representations: manual and automatic transcripts and their summaries.
  - We present a study on how discrete and continuous text informativeness measures may be extended to passage interestingness evaluation. In this context, passage interestingness is defined as a generalization of informativeness, whereby the information need is diverse and formalized as an unknown set of implicit queries.

1.5 Structure of the Thesis

This PhD thesis explains in Chapter 2 essential concepts and the state-of-the-art related to the research done during the development of this thesis. Scientific contributions are presented from Chapter 3 to Chapter 8. Finally, general conclusions and future work are presented in Chapter 9. In detail, each chapter is organized as follows:

In Chapter 2 we first describe the basic concepts of Neural Networks (NNs) and text representation. We then present the state-of-the-art concerning Sentence Boundary Detection (SBD), Automatic Text Summarization (ATS) for Arabic, and audio and text-based multimedia summarization. Finally, we explain evaluation metrics related to SBD and ATS.

Chapter 3 is dedicated to SBD. We first present and discuss some state-of-the-art SBD systems. Then, we explain our contribution for SBD and its evaluation from a multilingual perspective. We next focus on SBD for Modern Standard Arabic (MSA) to study how tuning a big out-of-domain dataset with a smaller in-domain dataset may help improving general SBD performance. Finally we we study the impact that transcription errors have over SBD.

In Chapter 4 we tackle text-based multimedia summarization for MSA. We first introduce ARTEX for Modern Standard Arabic (ARTEX-MSA), an extension to Autre Résumeur de TEXtes (ARTEX) capable of generating extractive summaries in MSA. Then,

we perform an automatic evaluation of ARTEX-MSA over a controlled dataset. Finally, we conduct a study to evaluate the performance of ARTEX-MSA over MSA automatic transcripts.

Chapter 5 is committed to audio-based multimedia summarization. First we explain how the probability distribution divergence is used over an audio-based multimedia summarization framework and we describe in detail our summarization proposal. We then introduce a second audio summarizer based purely on audio features. In the last part of this chapter we present and discuss the results obtained of both summarization strategies.

In Chapter 6 we present Window-based Sentence Boundary Evaluation (WiSeBE), a semi-supervised evaluation protocol based on multi-reference (dis)agreement, which debates if evaluating an automatic SBD system against a unique reference is enough to conclude the performance of the system. Then, we evaluate two SBD systems following a multi-reference strategy, were we compare a standard SBD evaluation against WiSeBE.

In Chapter 7 we conduct a transcript informativeness study based on ATS. We first present the protocol we followed to evaluate the informativeness of automatic transcripts and to measure the impact of ATS over informativeness. The, we present the results three evaluation scenarios, followed by a discussion of the relationship between evaluation metrics and informativeness.

In Chapter 8 we present an study on how discreet and continuous text informativeness measures may be extended to passage interestingness evaluation. We first explain our methodology to extend text informativeness measures to passage interestingness evaluation. Then, we present the obtained results over both informativeness and interestingness over two experimental scenarios.

In Chapter 9 we explain the final conclusions of this thesis and future work we want to achieve. We also present the advantages and limitations of the systems and proposal we have developed.

Chapter 2

State-of-the-art

Contents

During Chapter 1 we addressed the importance of automatic multimedia summarization on helping the understanding of the big amount of information that is now accessible thanks to online availability. This Chapter is focused on presenting the grounds of automatic multimedia summarization and related topics. In sections 2.1 and 2.2 we first describe basic and relevant concepts related to text representation and Neural Networks (NNs). This two sections may help the reader familiarizing with some concepts that are used all along this PhD thesis. Then, from Section 2.3 to 2.5, we present recent research that has been done regarding Sentence Boundary Detection (SBD), Automatic Text Summarization (ATS) for Arabic, and audio and text-based multimedia summarization. Finally, in Section 2.6 we address how SBD and ATS are normally evaluated.

2.1 Document and Word Representation for Natural Language Processing

Before any Natural Language Processing (NLP) analysis is performed over a set of documents, it is necessary to represent them into a suitable format able to capture the relation between the elements inside those documents. A Vector Space Model (VSM) (Salton et al., 1975), also known as bag-of-words, is a way of representing the relation between a set of documents $D = \{d_1, d_2, \ldots, d_M\}$ and a set of characteristics $V = \{v_1, v_2, \ldots, v_N\}$ intrinsic to the documents. The set of characteristics V may represent words, stems, letters, or other text unit. This kind of text representation has the particularity of not paying attention to the relation between the terms of a document.

A VSM represents the relation document $\leftrightarrow$ characteristic by the matrix $\mathbf{S}^{[M \times N]}$ defined as:

$$\mathbf{S} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ s_{2,1} & s_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ s_{M,1} & s_{M,2} & \cdots & s_{M,N} \end{bmatrix} \tag{2.1}$$

where each element $s_{i,j}$ corresponds to the relation between the characteristic v_j and the document d_i. $s_{i,j}$ may be represented as a binary presence/absence interaction of the characteristic v_j in the document d_i as follows:

$$s_{i,j} = \begin{cases} 1 & \text{if } v_j \in d_i \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Another possibility is to represent $s_{i,j}$ as the frequency of the characteristic v_j in the document d_i; however, a weighted frequency representation is a better option to avoid a voluminous, sparse and extremely noisy matrix (Torres-Moreno, 2014). The most popular weighted representation is tf.idf (Term Frequency, Inverse Document Frequency) (Spärck Jones, 1972). We define tf.idf in terms of D, V and $\mathbf{S}$ as:

$$s_{i,j} = tf.idf_{i,j} = C_{v_j}^{d_i} \cdot log_2 \left(\frac{|D|}{\sum_{d_i \in D} 1 \quad [v_j \in d_i]} \right) \tag{2.3}$$

where $C_{v_j}^{d_i}$ refers to the frequency of v_j in d_i. Different variants of tf.idf have been proposed taking into account a normalization term over the term frequency and the number of documents containing the term (Amini and Gaussier, 2013).

Word representation is another important aspect to take into account. Depending of the NLPs task to perform, representing the words within a dataset may be relevant in

Index	Word	One-hot Vector
1	tiger	[1 0 0 0 0 . . . 0 0]
2	cat	[0 1 0 0 0 . . . 0 0]
⋮	⋮	⋮
9 999	wolf	[0 0 0 0 0 . . . 1 0]
10 000	dog	[0 0 0 0 0 . . . 0 1]

Table 2.1: *One-hot Encoding Vectors Example*

terms of performance, usability and complexity. A first approach is to represent the words in form of categorical variables (one-hot encoding). The second approach is to create a continuous representation (word embeddings) of them.

2.1.1 One-hot Encoding

One-hot Encoding represents a dataset in the form of categorical variables. In this type of encoding, each word is characterized by a unique one-hot binary vector that is absolutely independent of the rest of the words in the document. For a dataset containing a vocabulary $V = \{v_1, v_2, \ldots, v_N\}$, each word v_i is represented by a N-dimension one-hot vector $\mathbf{v}_i = \langle c_1, c_2, \ldots, c_N \rangle$, where each component

$$c_j = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases}$$

Using one-hot encoding to represent a dataset is straightforward and well suited for some NLP tasks; however, some limitations are present. Table 2.1 exemplifies the one-hot vectors of a fictitious dataset composed of a vocabulary V of 10 000 words. Even the existent relation between $tiger \leftrightarrow cat$ and $wolf \leftrightarrow dog$, their corresponding one-hot vectors are incapable of maintaining it. Vocabulary size is another restraint; dimension of one-hot vectors depends directly of the number of words in the dataset. A dataset with one million different words will be represented by one million different vectors of one million components each, producing a $M^{[1000000 \times 1000000]}$ diagonal matrix. Given that the M matrix is highly related to its source dataset, reusability is another limitation.

2.1.2 Word Embeddings

Discrete text representations like one-hot encoding transform the words of a dataset into a bag-of-words space that leads to large sparse vectors and semantic information loss. Continuous text representations or word embeddings are capable of overcoming these limitations by representing the dataset using a continuous subspace approach with a defined number of components or dimension. In this representation, vectors are capable of maintaining the relation between words in the dataset following Harris (1954)'s distributional semantics hypothesis: *"words that appear in the same contexts share*

similar meanings". Words that appear frequently in similar contexts are closer to each other in the embeddings space (Goodfellow et al., 2016). Different word embeddings representations have been proposed in literature (Deerwester et al., 1990; Blei et al., 2003; Pennington et al., 2014); however, for the purpose of this thesis we focus on a pair of approaches based on Neural Networks (NNs).

Word2vec

Word2vec (Mikolov et al., 2013a) is a popular NN embeddings model based on Mikolov et al. (2013c). It aims to map the vocabulary of a dataset into a multidimensional vector space in which the distance between the projections corresponds to the semantic similarity between them (Ng and Abrecht, 2015). Word2vec consists of two different model approaches for computing continuous vector representations of words from very large datasets: continuous bag-of-words (CBOW) model and continuous skip-gram model.

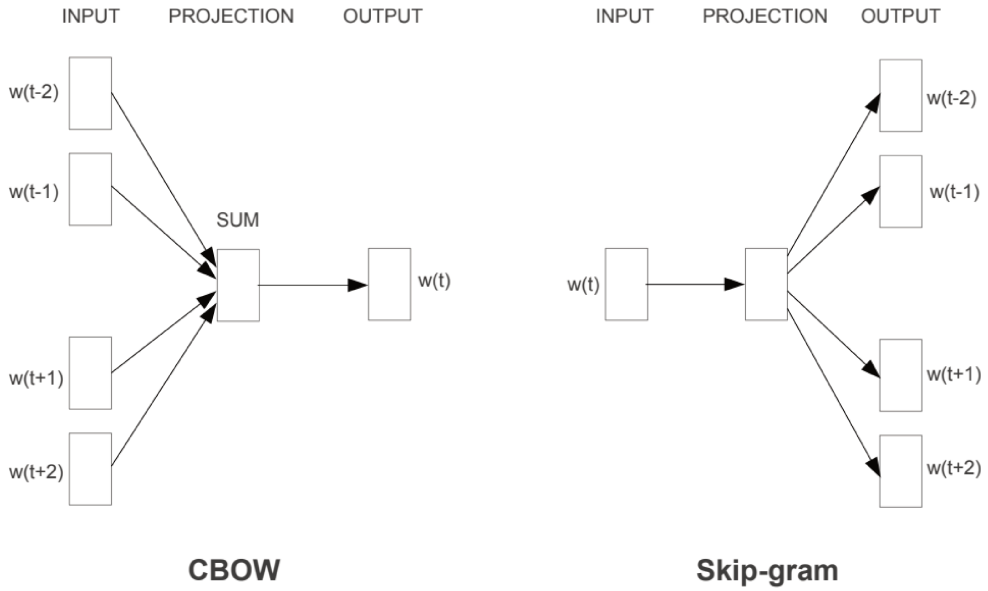

Figure 2.1: *Continuous Bag-of-words and Continuous Skip-gram Models (Mikolov et al., 2013a)*

CBOW model aims to predict a target word based on the given context, while the continuous skip-gram model aims to predict a target context given a word (Figure 2.1). In both cases the size of the embeddings is defined by the size of the projection layer. The ability of the continuous skip-gram model to automatically organize concepts and learn implicitly the relationships between them is exemplified in Figure 2.2.

Word embeddings produce by both CBOW and continuous skip-grams models may be trained with a very big dataset in the order of millions of words. Once trained, the resulting model may be used to vectorize a target dataset. However, some words in the new dataset may be unknown to the trained model. A simple solution to this inconvenient is to assign a fixed vector to all out-of-vocabulary (OOV) words. A more complex

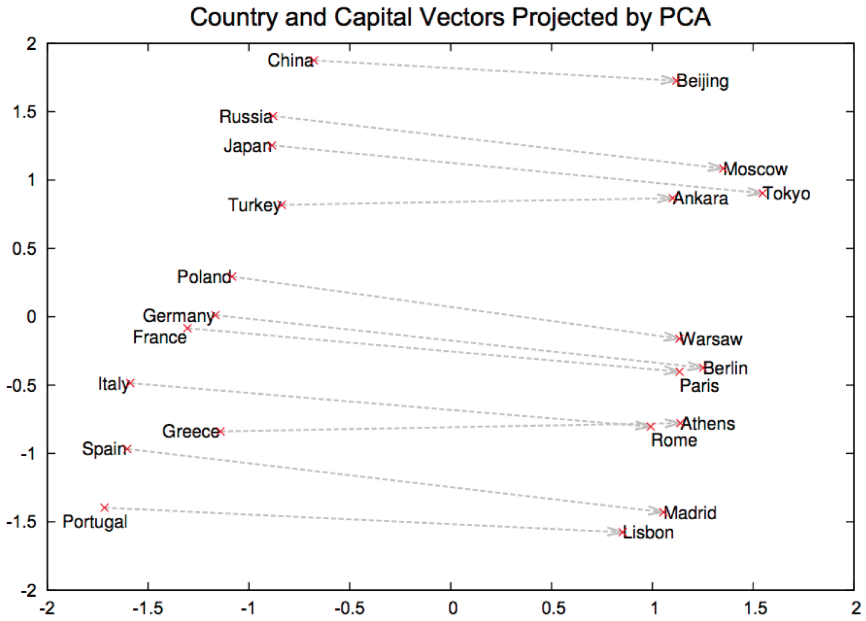

Figure 2.2: *Two-dimensional PCA Projection of the 1 000-dimensional Skip-gram Vectors of Countries and their Capital Cities (Mikolov et al., 2013b)*

solution was introduced by Bojanowski et al. (2016). They proposed the use of subword level information vectors to create word embeddings formed of word n-grams. In this manner, each word is expressed as the sum its n-grams; thus, the embedding of a new unknown word may be constructed by computing the average of the embeddings of its n-grams.

Fasttext

Fasttext[1] is a continuous text representation based on the skip-gram model (Mikolov et al., 2013a) proposed by (Bojanowski et al., 2016). In this approach, each word is represented as a bag of character n-grams and a vector is associated to each character n-gram; thus, word vectors are represented as the sum of their n-gram vectors.

In the original skip-gram model, given the training corpus $W = \{w_1, w_2, \ldots, w_T\}$, the objective is to maximize the log-likelihood defined as:

$$\sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c | w_t) \tag{2.4}$$

where C_t refers to the set of context words surrounding the word w_t. The simplest way to define the probability of the context word w_c given w_t is by computing the softmax function defined as:

[1] `https://fasttext.cc/`

$$\text{softmax}(s(w_c|w_t)) = p(w_c|w_t) = \frac{e^{s(w_t,w_c)}}{\sum_{j=1}^{T} e^{s(w_t,w_j)}} \tag{2.5}$$

where $s(w_t, w_c) = \mathbf{u}_{w_t}^{\top} \cdot \mathbf{v}_{w_c}$. Vectors $\mathbf{u}_{w_t}$ and $\mathbf{v}_{w_c}$ correspond to the embeddings of words w_t and w_c respectively.

In the Fasttext model, a different scoring function s is proposed based on subword vectors. Given a word w_t composed by the set of n-grams G_{w_t}, a vector representation $\mathbf{z}_{w_t}$ is associated to each n-gram $g \in G_{w_t}$. Thus the scoring function is defined as:

$$s(w_t, w_c) = \sum_{g \in G_{w_t}} \mathbf{z}_g^{\top} \cdot \mathbf{v}_{w_c} \tag{2.6}$$

This scoring function allows sharing the representations across words, and allows to learn representation for rare and unknown words.

2.2 Artificial Neural Networks

A biological neuron is a highly specialized electrically excitable cell (Figure 2.3). It is composed of the soma, the axon and the dendrites. The soma corresponds to the neuron's core which contains genetic information, maintains the neuron's structure, and provides energy to the rest of the cell. The dendrites are a root structure that branch out from the soma. Soma and dendrites receive electrical signals from the axons of other networks. The axon is a tail-like structure which emerges from the soma and conducts electrical signal through other neurons by the axon terminals, where synapses are located. The axon of a typical neuron has a few thousand synapses with other neurons (Hertz, 2018).

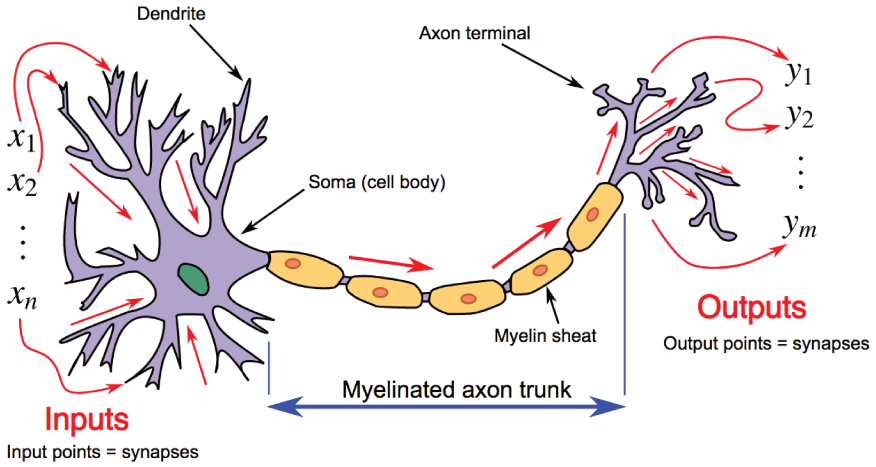

Figure 2.3: *Simplified Scheme of a Biological Neuron (Vu-Quoc, Loc, 2016)*

From hook-and-loop fasteners (better known as velcro) to bullet trains, a lot of human inventions have been the result of nature and biological inspiration. A good example of this is the perceptron or artificial neuron, based on biological neurons. Perceptrons established the basis for actual Artificial Neural Networks (ANNs) (or just Neural Networks (NNs)), capable of creating new Rembrandt paintings[2], helping doctors with cancer treatment options[3] and translating between more than 10 000 languages[4].

In 1943, inspired by nervous systems and neural events, neurophysiologist Warren McCulloch and mathematician Walter Pitts created the first mathematical model of a neural network, which aimed to mimic human thinking processes (McCulloch and Pitts, 1943). Psychologist Donald Olding Hebb, based on the mechanism of neural plasticity, proposed in 1949 that learning occurs in the brain primarily through the formation and change of synapses between neurons (Hebb, 1949). Finally, in 1958, psychologist Frank Rosenblatt proposed a more general model than McCulloch–Pitts's with Hebb's postulate: the perceptron.

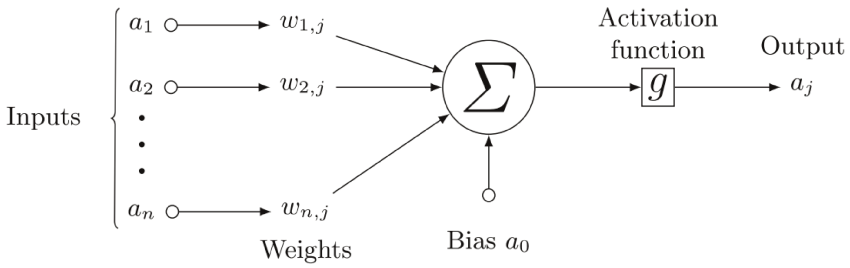

Figure 2.4: *General Model of an Artificial Neuron (Holzinger et al., 2017)*

The perceptron is an abstraction of a biological neuron. Figure 2.4 illustrates this abstraction, where each dendrite is represented by an input-weight pair, the soma as a transfer function with an activation function g and the axon as the result of the linear combination of all $w_{i,j} \cdot a_i$ pairs which is then fed into the activation function g. The output is estimated as follows:

$$a_j = g\left(\sum_{i=0}^{n} w_{i,j} \cdot a_i\right) \tag{2.7}$$

The goal of the activation function g is to introduce non-linearity into the model and provide a smooth transition as input values change. Most used activation functions are:

- Linear function: $g(x) = x$
- Rectified Linear Unit (ReLU) function: $g(x) = max(0, x)$
- Sigmoid function: $g(x) = \dfrac{1}{1 + e^x}$

[2] https://www.nextrembrandt.com/

[3] https://www.ibm.com/us-en/marketplace/clinical-decision-support-oncology

[4] https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html

- Hyperbolic tangent function: $g(x) = tanh(x)$

Learning

The artificial neuron learns by adapting each weight $w_{i,j}$ of the neuron j with an optimization algorithm. It follows the Hebb's rule introduced in Hebb (1949): *"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."* In the artificial neuron, the growing process is simulated modifying the artificial neuron's weight by minimizing the difference between expected and calculated values. The combination of weights which minimizes the error function is considered to be a solution of the learning problem.

For a NN, the learning process is driven by a process called back-propagation. It follows a gradient-based optimization algorithm that uses the chain rule to minimize the global error from the network by propagating the gradient from the output layer through the hidden layers up to the input layer. The main feature of back-propagation is its iterative, recursive and efficient method for calculating the weights updates to improve the network until it is able to perform the task for which it is being trained (Rojas, 2013; Goodfellow et al., 2016).

A NN is classified depending of how its neurons are interconnected. In the following paragraphs we describe the two types of NNs that are relevant to this thesis.

Feedforward Neural Networks

Feedforward Neural Networks (FFNNs), also known as multilayer perceptrons, are a kind of NNs in which information flows from the input layer to the output layer, passing through intermediate hidden layers without any loop. An example of a simple FFNN architecture is shown in Figure 2.5. It is composed of one input layer with three different features (V_0), two hidden layers with five neurons each (V_1, V_2) an output layer (V_3) with three output neurons.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a kind of NN for processing data that can be represented in a grid topology like time-series, pixels or word sliding windows. Originally developed for image processing (LeCun et al., 1989), CNNs employ a specialized kind of linear mathematical operation called convolution in place of general matrix multiplication in at least one of their layers (Goodfellow et al., 2016).

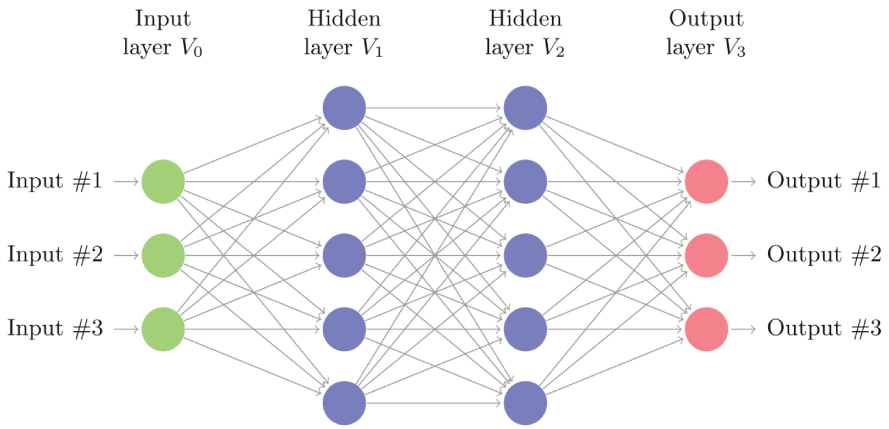

Figure 2.5: *Simple Feedforward Neural Network Architecture (Holzinger et al., 2017)*

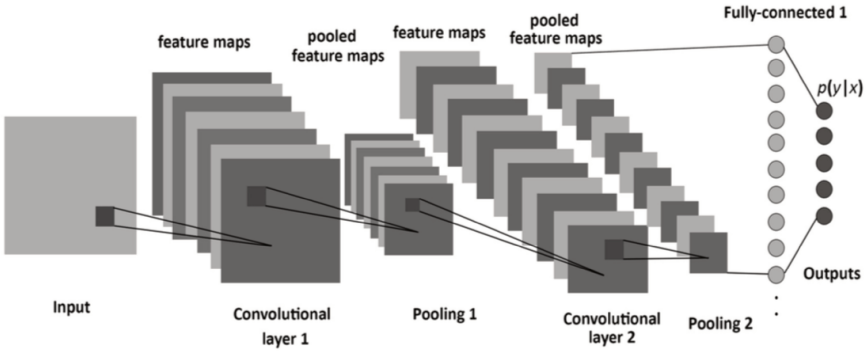

Figure 2.6: *Convolutional Neural Network Architecture (Albelwi and Mahmood, 2017)*

As seen in Figure 2.6, a CNN is built of convolution, pooling and fully-connected layers. The convolution layers perform a dot product between two matrices, where one matrix is the set of learnable parameters (kernel), and the other matrix is the restricted portion of the receptive field. The pooling or down-sampling layers aim to reduce the resolution of the feature maps produced by the convolution layers. They replace the output of the NN at a certain location with a summary statistic of the nearby outputs. Finally, the fully connected layers work as a FFNN. They extract the global features created by the convolutional and pooling layers (Albelwi and Mahmood, 2017).

2.3 Sentence Boundary Detection

Sentence Boundary Detection (SBD) aims to divide into segments a manual or automatic transcript. It has been a major research topic since Automatic Speech Recognition (ASR) moved to more general domains as conversational speech (Meteer and Iyer, 1996; Shriberg and Stolcke, 1996; Stolcke and Shriberg, 1996). Performance of ASR systems has improved over the years with the inclusion and combination of new Neural

Speech Transcript	SBD Applied to Transcript
two two women can look out after a kid so bad as a man and a woman can so you can have a you can have a mother and a father that that still don't do right with the kid and you can have to men that can so as long as the love each other as long as they love each other it doesn't matter	two // two women can look out after a kid so bad as a man and a woman can // so you can have a // you can have a mother and a father that // that still don't do right with the kid and you can have to men that can // so as long as the love each other // as long as they love each other it doesn't matter //

Table 2.2: *Sentence Boundary Detection Example*

Networks (NNs) methods (Hinton et al., 2012; Yu and Deng, 2016; Fohr et al., 2017). However, as a general rule, the output of ASR systems lacks of any syntactic information such as capitalization and sentence boundaries, showing the interest of ASR systems to obtain the correct sequence of words with almost no concern of the overall structure of the document (Gotoh and Renals, 2000).

Similar to SBD is the Punctuation Marks Disambiguation (PMD) task. PMD goal is to segment a formal written text into well formed sentences based on the existent punctuation marks (Palmer and Hearst, 1994, 1997; Kiss and Strunk, 2006; Treviso et al., 2017). In this context a sentence is defined (for English) by the Cambridge Dictionary[5] as:

> *"a group of words, usually containing a verb, that expresses a thought in the form of a statement, question, instruction, or exclamation and starts with a capital letter when written"*.

PMD carries certain complications, some given the ambiguity of punctuation marks within a sentence. A period can denote an acronym, an abbreviation, the end of the sentence or a combination of them as in the following example:

> *The U.S. president, Mr. Donald Trump, is meeting with the F.B.I. director Christopher A. Wray next Thursday at 8p.m..*

Despite these difficulties, PMD profits of morphological and lexical information to achieve a correct sentence segmentation. By contrast, segmenting an ASR transcript should be done without any (or almost any) lexical information and a flurry definition of sentence. The obvious division in spoken language may be considered speaker utterances. Nevertheless, in a normal conversation or even in a monologue, the way ideas are organized differs largely from a written text. These differences, added to disfluencies like revisions, repetitions, restarts, interruptions and hesitations make the definition of a sentence unclear, thus complicating the segmentation task (Strassel, 2003). Table 2.2 exemplifies some difficulties that are present when working with spoken language documents.

[5] `https://dictionary.cambridge.org/`

Even if there is no a general consensus regarding what a segment is, different methods have been studied to perform SBD over automatic and manual transcripts. The vast majority of them have focused on Language Models (LMs), classification, machine translation and NN approaches. English has been the most studied language in the context of SBD (Meteer and Iyer, 1996; Stolcke and Shriberg, 1996; Mrozinski et al., 2006; Liu et al., 2006; Lu and Ng, 2010; Nicola et al., 2013; Che et al., 2016b,a); nevertheless, research has also been done for French (Kolář and Lamel, 2012; Peitz et al., 2014), German (Peitz et al., 2014), Hungarian (Szaszák and Beke, 2012; Szaszák et al., 2016) and Arabic (Zribi et al., 2016). In the following paragraphs we present some of these approaches.

Meteer and Iyer (1996) divided speaker utterances into segments consisting each of a single independent clause. They considered a segment to begin either at the beginning of an utterance or after the end of the preceding segment. Any dysfluency between the end of the previous segment and the begging of current one was considered part of the current segment. They trained a 3-gram language model with a Good-Turing back-off mechanism for smoothing unseen n-gram estimates. For this, they considered that any word had two possible paths: 1) a transition to the next word or 2) a transition to the next word through a segment boundary. At every point they picked the most likely path as the history for the next word.

Stolcke and Shriberg (1996) considered the following set of linguistic structures as segments:

- Complete sentences
- Stand-alone sentences
- Disfluent sentences aborted in mid-utterance
- Interjections
- Back-channel responses

They constructed a language model where they associated a state depending if the word corresponded to a segment starting (**<S>**) or not (**<NO-S>**). During testing, they ran a hidden segment model which hypothesized segment boundaries between any two words. Finally, they implemented the Viterbi algorithm to find the most likely sequence of **<S>** and **<NO-S>** for a given word string. Based on this research, Mrozinski et al. (2006) trained two word-based and a class-based language models which then they combined by linear interpolation to compute the final probabilities.

Liu et al. (2006) drove a wide study on one of the main problems of SBD: imbalanced data. They focused on how imbalanced data samples impacts the segmentation process. For this study they constructed a Hidden Markov Model (HMM) to detect sentence boundaries using prosodic and textual information; and tested a variety of sampling approaches and a bagging scheme. During this study they followed the segmentation scheme proposed by the Linguistic Data Consortium[6] on the Simple Metadata Annotation Specification V5.0 guideline (SimpleMDE_V5.0) (Strassel, 2003), dividing

[6] `https://www.ldc.upenn.edu/`

the transcripts in Semantic Units (SUs). A SU seems to be an inclusive concept of a segment and is flexible enough to deal with the majority of spoken language troubles. For the purposes of this thesis we adopt the SU concept to define a segment.

Textual and acoustic features were also used by Kolář and Lamel (2012) to segment and predict punctuation marks. They created a statistical punctuation model based on an adaptive boosting mechanism to combine a set of learning algorithms to produce an accurate classifier.

Lu and Ng (2010) and Nicola et al. (2013) focused on conversational speech to create a SBD system based on dynamic conditional random fields relying only on textual information. Added to segmenting the transcript, they aimed to predict different punctuation marks like comma, period, question and exclamation marks. Peitz et al. (2014) implemented a hierarchical phrase-based translation approach based also on textual features to treat SBD as a translation task.

Szaszák and Beke (2012) and Szaszák et al. (2016) created a speech prosody based tokenization method to segment an audio in phonological phrases. They described a phonological phrase as a prosodic unit of speech, characterized by a single stress and that often corresponds to a group of words belonging syntactically together. They used audio features such as fundamental frequency of speech (F0) and speech signal energy. They based the segmentation system on HMMs, where a Viterbi alignment was applied in segmentation mode to recover the most likely underlying phonological phrase structure.

Zribi et al. (2016) proposed a pair of rule-based and statistical-based approaches for SBD of Tunisian Arabic using textual and audio features. For the rule-based method they established a set of had-made rules based on punctuation marks, conjunctions and other connectors. Regarding the statistical method they treated SBD as classification problem and proposed to classify each word in one of the following classes: 1) first word of a sentence, 2) word inside a sentence, 3) last word of a sentence and 4) one word sentence. Added to this two methods they proposed a hybrid method combining the rule and statistical methods.

NNs were implemented with word embeddings by Che et al. (2016b) to perform SBD and predict commas, periods and questions marks using only textual features. They explored three different NN models; the first one consisted on a Feedforward Neural Network (FFNN) architecture, while the two others followed a Convolutional Neural Network (CNN) one. Concerning the word embeddings used to perform experiments, Che et al. (2016b) opted for pre-trained GloVe[7] word vectors (Pennington et al., 2014). This kind of word embeddings use a distinct vector representation for each word ignoring the morphology of words. Che et al. (2016a) recovered the standard FFNN architecture presented in Che et al. (2016b) and introduced an acoustic model in a 2-stage joint decision scheme to predict the sentence boundary positions.

Based on the scheme described in Che et al. (2016b), in Chapter 3 we approach SBD as a binary classification task with CNNs to predict if a target word inside a sliding

[7] https://nlp.stanford.edu/projects/glove/

window corresponds to a boundary (**<SEG>**) or not (**<NO SEG>**). Different from GloVe vectors we opted for Fasttext embeddings, which take into account the morphology and make possible to represent unknown words.

2.4 Automatic Text Summarization for Arabic

In Section 1.1 we reviewed the concept of *summary* and the six elements we identified all summaries contemplate. One of these elements is the method that was followed to create the summary. It is possible to classify the method into extractive, abstractive and compressive, depending of the type summary it produces (Mani and Maybury, 2001; Torres-Moreno, 2014).

- **Extractive**: This type of method produces summaries which content is a set of sentences from the source document. Most relevant sentences are selected depending a heuristic or scoring function.
- **Abstractive**: In this case, summaries are produced by reformulating sentences from their source documents; resulting in summaries in which some of the content is not present in the corresponding source. Reformulation is done by rewriting and/or paraphrasing the text.
- **Compressive**: This type of method creates summaries that contain the same number of sentences as their sources. The length of the sentences is shorter given a shrinking process that is performed to preserve only the main information.

Automatic Text Summarization (ATS) has been an active field for the last 50 years and has shown a constant improvement. However, the research done for summarizing Arabic documents is recent and has been growing slowly compared to other languages such as English (El-Haj et al., 2011). To our knowledge, the Document Evaluation Conference 2004 (DUC2004)[8] was the first evaluation conference that considered Arabic as one of the languages to process. The DUC2004 included five tasks, two of which addressed the task of cross-lingual summarization between Arabic and English.

Given the complexity of Arabic in terms of morphology and structure, ATS for Arabic is more challenging than other languages (Al-Saleh and Menai, 2016). Al Qassem et al. (2017) presented an extended survey of methodologies and systems of ATS systems for Arabic that deal with these challenges. In the following paragraphs we describe some of the systems that have been created in the context of ATS for Arabic.

Azmi and Al-Thanyyan (2012) developed an extractive summarization approach in which no learning process is needed. The approach consists of a two-pass algorithm which first step is in charge of selecting a set of candidate sentences using rhetorical structure theory. During the second step, each candidate sentence is first passed through an important preprocessing phase consisting of sentence and word segmentation, stop-word elimination and root extraction. Then, each sentence is ranked using a

[8] `https://duc.nist.gov/duc2004/`

scoring scheme and the final summary is obtained with a dynamic programming technique which maximizes the general score of the summary with the constraint of not exceeding a given Compression Ratio (CR).

ESMAT, developed by Binwahlan (2015), is an extractive summarizer based on a linear combination of textual features. The summarizer is divided in two steps: 1) preprocessing and feature extraction, and 2) summary generation. The preprocessing and feature extraction step first uses line breaks to divide the document into sentences. Then, an stemming and stop words removal phase is performed. Finally, scores for the six following features are computed: average TF-ISF, sentence length, sentence position, sentence similarity to document, sentence concepts and log entropy. During the second step, the summary is generated by computing the average score of each sentence and selecting the ones that are best ranked.

Azmi and Altmami (2018) proposed a four phase abstractive summarizer with user controlled granularity. During the first phase of the summarizer, each document is segmented into topics with a variant of ArabTiling (Harrag et al., 2010), a topic segmentation system for Arabic based on Textling (Hearst, 1997). Then, a headline is constructed for each topic by computing the tf.idf score of each word and selecting the four best ranked words. During the third phase, a generic extractive summary is generated for each topic. The method that is followed to create this summary is based on Azmi and Al-Thanyyan (2012). Finally, the fourth and last phase generates the abstractive summary based on the summary obtained in the previous phase. This phase follows a sentence reduction scheme that removes extraneous words, sub sentences between some words and clauses.

Qaroush et al. (2019) proposed a set of extractive summarizers that evaluate each sentence of a document based on a combination of statistical and semantic features taking into account sentence importance, coverage and diversity. For all summarizers they followed a three phase methodology composed of 1) text preprocessing, 2) feature extraction, and 3) sentence evaluation and selection. The text preprocessing phase aims to transform the document into a unified representation. It performs the following operations: tokenization, letters normalization, stop-words removal and stemming. Concerning the feature extraction phase, it identifies a set of statistical and semantic features including key phrases, sentence location, title similarity, sentence length, cue words, numbers and emphasizing words. Finally, the sentence evaluation and selection phase takes into account the scores of the extracted features to produce the summary. During this phase two types of summarization techniques are possible: score-based and supervised machine learning. The score-based technique simply computes a linear combination of all score features while the supervised machine learning technique trains a binary classifier to decide if a sentence should be or not in the final summary.

2.5 Audio and Text-based Multimedia Summarization

In this section we will revise another of the six elements all summaries contemplate: the source. In a multimedia context, the source type of a document to be summarize does not limit to text uniquely. Depending of the source type the summarization process is conducted different. If an audio signal is considered as a source, it is possible to drive the summarization process in three different ways: 1) using only the audio features and perform summarization with audio techniques, 2) obtaining a text representation of the audio signal and use textual summarization techniques to get the most relevant parts of the source or 3) perform a hybrid approach considering audio and text features/techniques. The rest of this section details each approach, as well as their advantages and disadvantages.

Audio-based Multimedia Summarization

Audio summarization without any textual representation aims to produce an abridged and informative version of an audio source using only the information contained in the audio signal. This kind of summarization is challenging because the summary creation is driven only on how things are said; however, it is advantageous in terms of transcripts availability.

Maskey and Hirschberg (2006) presented an audio-based summarization method based on a Hidden Markov Model (HMM) framework. The method uses the 12 following normalized acoustic features to represent the HMM observation vectors: speaking rate; F0 min, max, mean; F0 range and slope; min, max and mean Root Mean Square (RMS) energy; RMS slope and sentence duration. The inclusion or exclusion of the segment within the summary represents the hidden states of the HMM. The method uses the Viterbi algorithm to find the optimal sequence of sentences to include in the summary. Finally, the selected sentences are concatenated to form the summary.

Duxans et al. (2009) developed an audio-based summarizer for soccer games that detects highlighted events based on two acoustic features of the soccer game audio track: block energy and acoustic repetition index. The block energy feature aims to capture the audio energy evolution of the audio track to discriminate between hot spots and neutral moments. The acoustic repetition index feature relies on the observation that during the hot spots of a match (or just before/after them), it is very common for sport commentators to repeat or to lengthen vowels in certain words such as “goal”, the name and nickname of a player. It represents the correlation between a narrow acoustic section and the seconds just before/after it. After the acoustic features are computed, the system lists all hot spots positions and their score, indicating the relevance of each of them. The summary is created by concatenating the video clips of the most relevant hot spots.

Zlatintsi et al. (2012) addressed the audio-based summarization task by exploring the potential of a modulation model for the detection of perceptually important audio events. They performed a saliency computation of audio streams based on a set of

saliency models and various linear, adaptive and nonlinear fusion schemes. They approached the saliency computation as a problem of assigning a measure of interest to audio frames, based on spectro-temporal cues. This resulted in a compact representation of the audio stream by tracking the components with maximal energy contribution across frequencies and time. The summary was created by selecting the most salient frames until the desired Compression Ratio (CR) was reached.

Text-based Multimedia Summarization

Directing the audio summary with textual methods benefits from the information contained within the text, dealing to more informative summaries. Text-based multimedia summarization needs automatic or manual speech transcripts to select the pertinent segments and produce an informative summary. Nevertheless, speech transcripts may be expensive, non available or of low quality; which impacts the summarization performance.

Taskiran et al. (2006) proposed a text-based audio summarizer method based on word frequencies scoring, word co-occurrence detection (collocations) and coverage maximization. The method first divide the transcript into segments based on pauses detection. Next, a term frequency score (based on tf.idf) is computed for each segment after a stop words filtration process. Then, collocations are detected by computing the log likelihood between word bi-grams. Finally, the summary is generated by selecting the segments with the highest score to duration ratios while at the same time maximizing the coverage of the summary over the full transcript.

Christensen et al. (2008) developed a text-based audio summarizer conformed of three cascading phases: 1) converting the audio stream into text using an Automatic Speech Recognition (ASR) system, 2) segmenting the transcript into utterances and stories, and 3) determining which utterances should be highlighted using a saliency score. Once the ASR system generates the corresponding transcript of the audio signal, the system performs utterance and stories segmentation using a maximum entropy model. Concerning utterances segmentation, the model provides statistics for assigning a probability to each word indicating to which degree it is the last word before an utterance boundary. Story segmentation follows the same principle with the difference that it operates on an utterance level. The saliency scoring phase aims to identify utterances with a high degree of information. Each utterance is then represented by four features: position, length, tf.idf and cosine similarity. This four features are the input of a Neural Network (NN) composed of two layers. The first layer processes individual features derived form each utterance while the second one combines the outputs of the first layer.

Rott and Červa (2016) created a text-based audio summarization system composed of three independent components: 1) an ASR system, 2) a syntactic analyzer and 3) a summarizer. After applying an ASR system based on NNs over the input audio, the resulting transcript is parsed into a phrasal tree using a syntactic analyzer. The summarizer component (Rott and Červa, 2013) consists of two modules: preprocessing and

summarization. During the preprocessing module, each phrase is lemmatized and all synonyms are normalized. Concerning the summarization module, it follows a tf.idf linear combination scoring technique to rate each segment based on its relevant vocabulary. The summary is created concatenating the top ranked segments until the CR is reached.

Audio+text based Multimedia Summarization

Using both audio features and textual methods can boost the summary quality; yet, disadvantages of both approaches are present.

Maskey and Hirschberg (2005) drove an empirical study of the usefulness of different types of features for extractive summarization. They evaluated lexical, prosodic, structural and discourse features as predictors of the segments which should be included in the summary or not using a Bayesian Network classifier. Regarding lexical features they considered counts of name entities types (persons, organizations and places) and number of words in the current, previous and following sentence. Similar to Maskey and Hirschberg (2006), they used speaking rate; F0 min, max, mean; F0 range and slope; min, max and mean RMS energy; RMS slope and sentence duration as prosodic features. Concerning structural features they opted for sentence position; previous, actual and next speaker type; speaker change and turn position. Respecting discourse features, they computed a diversity score that favored segments with new information while penalizing redundant information.

Zlatintsi et al. (2015) investigated the problem of audio salient event detection for extractive summarization, where audio saliency is assessed by auditory and perceptual cues such as Teager energy (Teager and Teager, 1990), loudness (Fastl and Zwicker, 2006) and roughness (Vassilakis, 2001). They followed a non-parametric data-driven classification approach with audio and text features based on a k-nearest neighbor classifier. They represented audio segments with 27 audio features along with its first and second temporal derivatives and four textual features including cosine similarity and lexical affective content.

Szaszák et al. (2016) proposed a summarization pipeline to produce extractive summaries. Different to previous works, segmentation is performed before any ASR process. The segmentation phase follows a speech prosody based tokenization to separate the audio signal into phonological phrases. It is based on a HMM which models each phonological phrase in terms of its F0 and signal energy. Once the phonological phrases are obtained, an ASR system produces the corresponding transcripts. The summarizer is divided in three modules; the first module is in charge of preprocessing the automatic transcripts by maintaining only nouns and performing stemming. The second module performs textual feature extraction; for each segment it computes the following textual features: tf.idf, Latent Semantic Analysis (LSA), and segment position and length. Finally, the third and last module selects the top ranked segments based on a linear combination of the textual features.

Audio summarization based only on acoustic features has the big advantage that no textual information is needed. This approach is especially useful when human transcripts are not available for the spoken documents or ASR transcripts have a high Word Error Rate (WER). However, for high informative contexts like documents from the *AMIS-Dataset* (Section 1.2.1), where most relevant information resides on the things that are said rather than on how they are said, a way to capture the informativeness that transcripts provide is needed. In Chapter 5 we introduce a hybrid approach during training phase while audio-based during summary creation.

2.6 Evaluation Measures

One of the objectives of this PhD thesis is to open the discussion about evaluation metrics and the subjectivity of gold standards in certain Natural Language Processing (NLP) tasks. In this Section we cover some evaluation measures related to Sentence Boundary Detection (SBD) and Automatic Text Summarization (ATS). In both cases subjectivity is an important factor where the following question arises: Is there a unique gold standard for a transcript segmentation or summary?

2.6.1 Sentence Boundary Detection Measures

As discussed in Section 2.3, SBD research has been focused on two different aspects: features and methods. Despite their differences in features and/or methodology, almost all works share a common element: the evaluation methodology. Measures as *Precision*, *Recall*, *F1-score*, Classification Error Rate (CER) and Slot Error Rate (SER) are normally used to evaluate the proposed system against a gold standard. A SBD system is normally the first step of a NLP pipeline, thus further tasks rely on the result of the SBD step; meaning that is crucial to have a good segmentation. But comparing the output of a system against a unique reference will provide a reliable score to decide if the system is good or bad? To our knowledge, the amount of studies that have tried to target the sentence boundary evaluation with a multi-reference approach is very small.

Bohac et al. (2012) compared the human ability to punctuate recognized spontaneous speech. They asked 10 annotators to punctuate about 30 minutes of Automatic Speech Recognition (ASR) transcripts in Czech. For an average of 3 962 words, the punctuation marks placed by annotators varied between 557 and 801; this means a difference of 244 segments for the same transcript. Over all annotators, the absolute consensus for period (.) was only 4.6% caused by the replacement of other punctuation marks as semicolons (;) and exclamation marks (!).

Kolář and Lamel (2012) considered two independent references to evaluate their system and proposed two approaches. The first one was to calculate the SER for each of one the two available references and then compute their mean. They found this approach to be very strict because for those boundaries where no agreement between references existed; the system was going to be partially wrong even the fact that it has

correctly predicted the boundary. Their second approach tried to moderate the number of unjust penalizations. For this case, a classification was considered incorrect only if it did not match either of the two references.

These two examples show the real need and some straightforward solutions for multi-reference SBD evaluation metrics. However, we think that it is possible to consider in a more inclusive approach the similarities and differences that multiple references could provide into a sentence boundary evaluation protocol. In Chapter 6 we introduce Window-based Sentence Boundary Evaluation (WiSeBE), a semi-supervised metric for evaluating SBD systems based on multi-reference (dis)agreement.

2.6.2 Automatic Text Summarization Measures

ROUGE

Lin (2004) introduced Recall-Oriented Understudy for Gisting Evaluation (ROUGE), a series of measures to determine the quality of an automatically generated summary by comparing it with a set of reference summaries created humans. The ROUGE measures count the n-grams intersection between the automatically generated summary to be evaluated and the reference summaries. The most common ROUGE measure is ROUGE-n, which is a n-gram recall between a candidate summary and the reference summary; n-grams are composed of n consecutive text units. In particular, ROUGE-1 computes the distribution of uni-grams; it counts the number of uni-grams (words, lemmas, stems, etc.) that occur both in the candidate and reference summaries (Torres-Moreno, 2014).

Let S be a candidate summary to be evaluated against a reference summary R, ROUGE-n is defined as:

$$\text{ROUGE-}n = \frac{\sum_{\omega \in R} count_{match}(\omega)}{\sum_{\omega \in R} count(\omega)} \tag{2.8}$$

where ω corresponds to a text unit (word, lemma, stem, etc.), $count(\omega)$ is the frequency of the ω-th n-gram in R and $count_{match}(\omega)$ is the co-occurring frequency of the ω-th n-gram in R and S.

While ROUGE-1 (uni-grams) considers each text unit independently, ROUGE-2 (bi-grams) considers sequences of two text units. Skip bi-grams, used in ROUGE-S, are any pair of text units in the sentence order with possible gaps between them; the number of skipped terms (gaps) is a parameter of ROUGE-S. An extension of ROUGE-S is ROUGE-SU, which expand ROUGE-S with the addition of uni-grams. ROUGE-2 and ROUGE-SU4 (skip bi-grams with a maximal gap size of 4 and uni-grams) were used to evaluate the generated summaries in the Document Understanding Conference (DUC) in 2005 (Dang, 2005). ROUGE proved itself better correlating human judgments under readability assumption than classical cosine measures (Radev et al., 2003). Indeed, the

various ROUGE variants were evaluated on three years of DUC data in Lin and Och (2004), showing that some ROUGE versions are more appropriate for specific contexts.

For cases where a very large number of documents have to be evaluated, reference summaries availability becomes a major issue. Thus, it becomes easier to apply a measure that can be used automatically to compare the content of the candidate summaries with the full set of source documents rather than comparing with human created reference summaries.

FRESA

FRamework for Evaluating Summaries Automatically (FRESA)[9] (Torres-Moreno et al., 2010; Saggion et al., 2010) is an automatic summary evaluation method inspired by the works of Lin et al. (2006) and Louis and Nenkova (2009) to evaluate summaries without the need of a reference. The method integrates a classic preprocessing step of the documents (stopwords filtering, normalization, etc.) before calculating the probability distribution divergences between the source document and the candidate summary. This preprocessing step allows to keep only the informative words and to focus on informativeness. FRESA allows to compute the probability distribution divergence with both Kullback-Leibler (KL) and Jensen-Shannon (JS) divergences. For two discrete probability distributions P and Q, the KL divergence D_{KL} of Q in relation to P is defined as:

$$D_{KL}(P||Q) = \frac{1}{2} \sum_{w \in P} \left(P_w \cdot log_2 \frac{P_w}{Q_w} \right), \tag{2.9}$$

$$P_w = \frac{C_w^T}{N_S + N_T} \quad \text{and} \quad Q_w = \begin{cases} \dfrac{C_w^S}{N_S} & \text{if} \quad w \in S \\ \dfrac{C_w^T + \delta}{(N_S + N_T) + \delta \cdot B} & \text{otherwise} \end{cases}$$

P corresponds to the probability distribution of words w in text T while Q corresponds to the probability distribution of words w in the summary S. N_T and N_S are the number of words in T and S respectively. $B = 1.5|V_T|$, where V_T is the vocabulary of T. C_w^T and C_w^S correspond the frequency of word w in T and S respectively. For smoothing the summary's probabilities δ is set to 0.005.

The JS divergence D_{JS} is the symmetrized version of D_{KL}. It is defined as:

$$D_{JS}(P||Q) = \frac{1}{2} \sum_{w \in P} \left(P_w \cdot \log_2 \frac{2P_w}{P_w + Q_w} + Q_w \cdot \log_2 \frac{2Q_w}{P_w + Q_w} \right) \tag{2.10}$$

with the same specification as for Equation 2.9.

[9] http://fresa.talne.eu

Pyramid

FRESA and all ROUGE variants base their functionality in the lexical similarities between the candidate summary and the reference (summary or source document). For abstractive or sentence compression summaries this is a big problem given the lexical disparity between source document and summary. Pyramid (Nenkova and Passonneau, 2004; Nenkova et al., 2007) is a semi-automatic protocol for evaluating content selection in summarization created by Nenkova and Passonneau (2004). It is based on the idea that there is not a unique summary from a collection of reference summaries that can be considered as the gold standard; but all summaries contribute equally with relevant information. The higher the agreement concerning a piece of information, the higher its information weight.

Given a set of reference summaries, units of meaning not bigger than a clause called Summary Content Units (SCUs) are identified by human evaluators. Each SCU has a weight corresponding to the number of summaries it appears in. The set of SCU are then ordered in a pyramidal shape, where each tier contains all and only the SCU with the same weight. The pyramidal shape is the result of a Zipfian distribution of the SCU weights; a few SCUs that all people express in their summaries are located at the top of the pyramid, while a very large number of SCUs, expressed by only one of the summary writers, form the base of the pyramid (Nenkova et al., 2007).

The weight D of a summary is given by the following equation:

$$D = \sum_{i=1}^{n} (i \times d_i) \tag{2.11}$$

where n is the number of reference summaries and d_i is the number of SCUs of the summary at a level i. The higher the weight of a summary, the better its quality.

2.7 Conclusion

In this chapter we discussed Sentence Boundary Detection (SBD), its difficulties and how some state-of-the-art methods have tried to overcome these difficulties. Independently of their performance, all revised methods focus on one language. In Chapter 3 we present a SBD system based on Convolutional Neural Networks (CNNs) and Fasttext vectors to tackle SBD in a multilingual context.

We also discussed the complexity of Automatic Text Summarization (ATS) for Arabic and its slow but constant evolution for the last 15 years. We examined relevant summarizing approaches, which most of them focalize in the preprocessing phase. In Chapter 4 we describe an extension to Autre Résumeur de TEXtes (ARTEX), a state-of-the-art extractive summarizer, able to create summaries for Modern Standard Arabic (MSA) in a lightweight and fast way.

Another relevant topic we covered in this Chapter was audio and text-based multimedia summarization. All the methods we described show advantages and disadvantages depending if they use only text or audio features, or a hybrid of both. In Chapter 5 we introduce a hybrid approach during training phase while text independent (audio-based) during summary creation.

Finally, we addressed the topic of evaluation metrics and their limitations when evaluating subjective tasks like SBD and ATS. In Chapter 6 we present a semi-supervised metric for evaluating SBD systems based on multi-reference (dis)agreement. Followed by Chapter 7, where we analyze the quality of Automatic Speech Recognition (ASR) systems from the perspective of ATS methods. Lastly, in Chapter 8 we study a set of informativeness measures and their ability to deal with interestingness evaluation.

Chapter 3

Sentence Boundary Detection

Contents

State-of-the-art Automatic Speech Recognition (ASR) systems produce good quality transcripts which may be used in further Natural Language Processing (NLP) tasks like Automatic Text Summarization (ATS), Machine Translation (MT), Question Answering (QA), etc. Nevertheless, ASR systems focus on obtaining the correct sequence of words with almost no concern of the overall structure of the documents, producing a lack of syntactic information. Sentence Boundary Detection (SBD) aims to restore part of the syntactic information separating the transcript into sentences. It may be conducted with two different types of features, and the selection of any depends of their availability and the methods that will be used.

The first type corresponds to acoustic features, which relies on the audio signal and the possible information that could be extracted like pauses, word duration, pitch and energy information (Kolář and Lamel, 2012; Igras and Ziółko, 2016; Che et al., 2016a). The second type, which is dependent of a manual or automatic transcription process,

corresponds to textual features like bag-of-words, word *n*-grams and word embeddings (Lu and Ng, 2010; Peitz et al., 2014; Che et al., 2016b; Zribi et al., 2016).

Given the spoken nature of the documents, SBD carries several complications, being the concept of sentence the principal one. Different from well formed written documents where ideas are appropriately organized in sentences, for spoken language the concept of sentence is much more fuzzy. The Linguistic Data Consortium[1] analyzed this problematic on the Simple Metadata Annotation Specification V5.0 guideline (SimpleMDE_V5.0) (Strassel, 2003) and suggested the use of sentence-like segments called Semantic Units (SUs). A SU is considered to be an atomic element of the transcript that manages to express a complete thought or idea on the part of the speaker. Sometimes, a SU corresponds to the equivalent of a sentence in written text, but other times (the most part of them) a SU corresponds to a phrase or a single word. SUs seem to be an inclusive conception of a segment, they embrace different previous segment definitions and are flexible enough to deal with the majority of spoken language troubles. For these reasons we will adopt the SU as our segment definition.

This chapter is organized as follows. In Section 3.1 we present a set of SBD systems based on Convolutional Neural Networks (CNNs) which we evaluate in a multilingual context. We then present an extended work regarding SBD for Modern Standard Arabic (MSA) in Section 3.2; where we conduct two experimental scenarios with two SBD systems based on Neural Networks (NNs) to study how tuning a big out-of-domain dataset with a smaller in-domain dataset may help to improve general SBD performance. Finally, in Section 3.3 we study the impact that transcription errors have over SBD.

3.1 Convolutional Neural Networks for Sentence Boundary Detection

Convolutional Neural Networks (CNNs) have shown to be useful for processing data that can be represented in a grid topology like time-series, pixels or word sliding windows. The best Sentence Boundary Detection (SBD) system reported by Che et al. (2016b) implemented a CNN to segment a transcript and predict commas, periods and questions marks using only textual features. Based on this work, we approach the SBD task as a binary classification problem to predict if the central word inside a sliding context window corresponds to a boundary (**<SEG>**) or not (**<NO SEG>**) using only lexical features. Each word in the sliding context window is represented by a Fasttext vector (Bojanowski et al., 2016), a type of word embedding with the particularity that each word is characterized as a bag-of-character *n*-grams and a vector representation is associated with each character *n*-gram; in this manner, each word is expressed as the sum its *n*-grams. We opted for a CNN based architecture given its capacity of establishing relationships between adjacent elements and extracting local features using a

[1] https://www.ldc.upenn.edu/

sliding window along an input matrix. Further explanation regarding CNNs is available in Section 2.2.

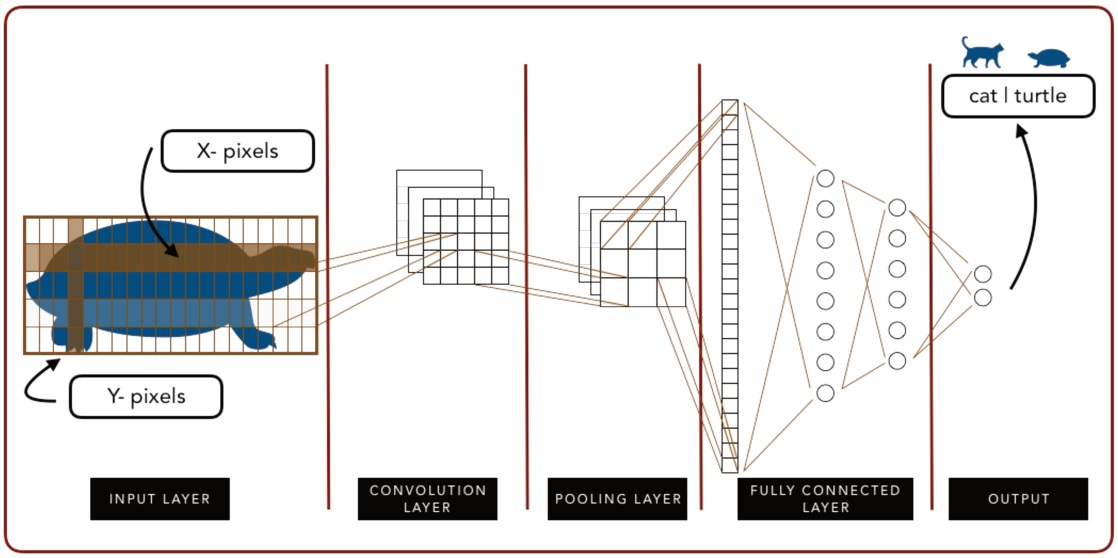

Figure 3.1: *CNN for Image Classification*

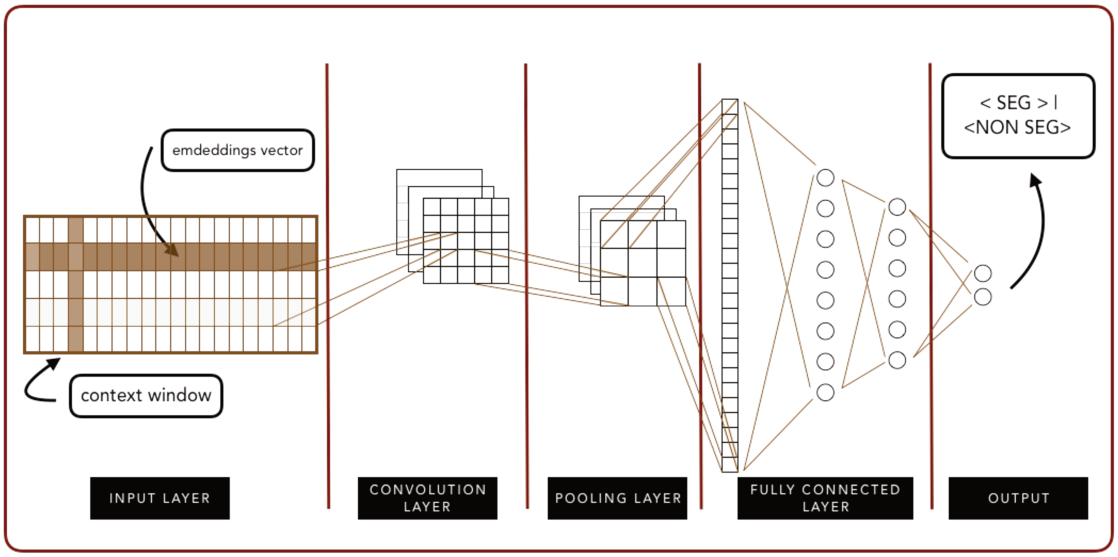

Figure 3.2: *CNN for Sentence Boundary Detection*

CNNs were originally created for computer vision tasks where the input layer corresponds to a matrix $\mathbf{X}^{[m \times n]}$ matrix and each cell $x_{i,j}$ is a pixel from the image (Figure 3.1). For the purpose of SBD, the input layer represents the relation between a context window X and their corresponding Fasttext vectors $\mathbf{V}$ (Figure 3.2). Given the intrinsic relation between the components of Fasttext vectors and the words within the sliding context window, it is feasible to make an extrapolation to the relation between adjacent pixels of an image. This way, the input matrix $\mathbf{X}^{[|X| \times dim(\mathbf{v})]}$ of the CNN is defined as

follows:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,dim(\mathbf{V})} \\ x_{2,1} & x_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{|W|,1} & x_{|W|,2} & \cdots & x_{|X|,dim(\mathbf{V})} \end{bmatrix} \tag{3.1}$$

where $|X|$ corresponds to the number of words in the sliding context window and $dim(\mathbf{V})$ to number of components of the Fasttext vectors.

For each context window X_t, the sliding step k_t depends of the predicted class for the context window X_{t-1}:

$$k_t = \begin{cases} 1 & \text{if } \textbf{<NO SEG>} \\ |X| \backslash 2 & \text{if } \textbf{<SEG>} \end{cases} \tag{3.2}$$

3.1.1 Convolutional Neural Networks Architectures

Che et al. (2016b) presented a pair of SBD systems based on CNNs which show interesting results over a dataset conformed by TED talks in English. Based on this architecture we propose the three different CNN architectures shown in Figure 3.3, where CNN-A is similar, in general terms, to Che et al. (2016b) best system. In CNN-B we reduce the complexity of the architecture by applying only two convolutional layers. Concerning CNN-C, only one fully connected layer is applied after the pooling layer.

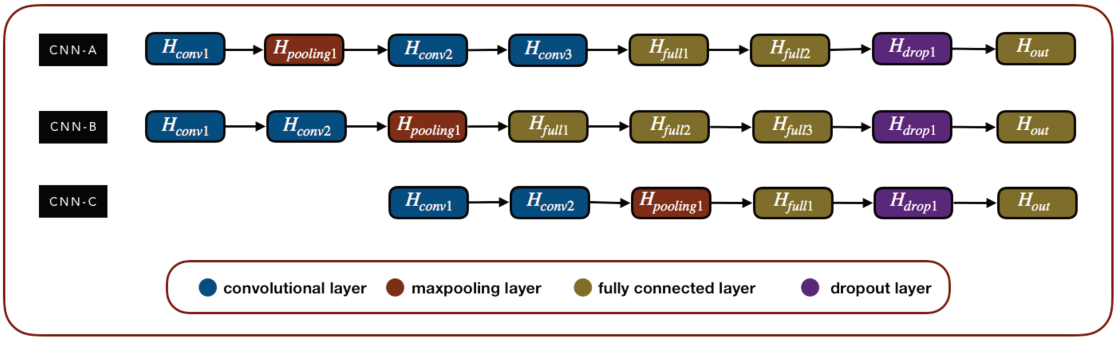

Figure 3.3: *CNN architectures for Sentence Boundary Detection*

CNN-A architecture is explained in Equations 3.3a to 3.3h. The first convolutional layer H_{conv1} is composed of a 2×4 kernel and 64 output neurons. The maxpooling layer $H_{pooling1}$ has a 2×3 kernel and a stride of 2×3. The second convolutional layer H_{conv2} has a 2×2 kernel and 128 output neurons, while H_{conv3} has a 1×49 kernel and 128 output neurons. All convolution layers have a stride of 1×1 and valid padding. The

fully connected layers H_{full1} and H_{full2} have 4 096 and 2 048 neurons respectively. The output of all convolutional, maxpooling and fully connected layers are in function of Rectified Linear Unit (ReLU) activations. The keeping probability of the dropout layer H_{drop1} is set to 80%. The output layer H_{out} corresponds to the classification layer with two neurons for **<SEG>** and **<NO SEG>**. A softmax activation function is applied in order to turns logits of H_{out} into probabilities that sum to one.

$$H_{conv1} = \text{relu}(\mathbf{X} \circledast W_{conv1} + b_{conv1}) \tag{3.3a}$$
$$H_{pooling1} = \text{maxpool}(H_{conv1}) \tag{3.3b}$$
$$H_{conv2} = \text{relu}(H_{conv1} \circledast W_{conv2} + b_{conv2}) \tag{3.3c}$$
$$H_{conv3} = \text{relu}(H_{conv2} \circledast W_{conv3} + b_{conv3}) \tag{3.3d}$$
$$H_{full1} = H_{conv3} \times W_{full1} + b_{full1} \tag{3.3e}$$
$$H_{full2} = H_{full1} \times W_{full2} + b_{full2} \tag{3.3f}$$
$$H_{drop1} = \text{dropout}(H_{full2}) \tag{3.3g}$$
$$H_{out} = \text{softmax}(H_{drop1} \times W_{out} + b_{out}) \tag{3.3h}$$

Architecture from CNN-B is presented in equations 3.4a to 3.4h. The first convolutional layer H_{conv1} is composed of a 3×3 kernel and 32 output neurons, while H_{conv2} has a 2×2 kernel and 64 output neurons. Both H_{conv1} and H_{conv2} have a stride of 1×1 and valid padding. The maxpooling layer $H_{pooling1}$ has a 2×3 kernel and a stride of 1×3. The fully connected layers H_{full1}, H_{full2} and H_{full3} have 2 048, 4 096 and 2 048 neurons each. The dropout (H_{drop1}) and output (H_{out}) layers are defined equal to CNN-A.

$$H_{conv1} = \text{relu}(\mathbf{X} \circledast W_{conv1} + b_{conv1}) \tag{3.4a}$$
$$H_{conv2} = \text{relu}(H_{conv1} \circledast W_{conv2} + b_{conv2}) \tag{3.4b}$$
$$H_{pooling1} = \text{maxpool}(H_{conv2}) \tag{3.4c}$$
$$H_{full1} = H_{pooling1} \times W_{full1} + b_{full1} \tag{3.4d}$$
$$H_{full2} = H_{full1} \times W_{full2} + b_{full2} \tag{3.4e}$$
$$H_{full3} = H_{full2} \times W_{full3} + b_{full3} \tag{3.4f}$$
$$H_{drop1} = \text{dropout}(H_{full3}) \tag{3.4g}$$
$$H_{out} = \text{softmax}(H_{drop1} \times W_{out} + b_{out}) \tag{3.4h}$$

CNN-C architecture is presented in equations 3.5a to 3.5f. Its configuration is the same as for CNN-B with the particularity that it has only one fully connected layer of 2 048 neurons between $H_{pooling1}$ and H_{drop1}.

$$H_{conv1} = \text{relu}(\mathbf{X} \circledast W_{conv1} + b_{conv1}) \tag{3.5a}$$
$$H_{conv2} = \text{relu}(H_{conv1} \circledast W_{conv2} + b_{conv2}) \tag{3.5b}$$
$$H_{pooling1} = \text{maxpool}(H_{conv2}) \tag{3.5c}$$
$$H_{full1} = H_{pooling1} \times W_{full1} + b_{full1} \tag{3.5d}$$
$$H_{drop1} = \text{dropout}(H_{full1}) \tag{3.5e}$$
$$H_{out} = \text{softmax}(H_{drop1} \times W_{out} + b_{out}) \tag{3.5f}$$

3.1.2 Experimental Evaluation

Experiments were performed with a subset of the French Gigaword First Edition[2] (Graff, 2006). This dataset is a comprehensive archive of newswire text data that has been acquired over several years by the Linguistic Data Consortium (LDC) at the University of Pennsylvania. It is composed of two international French newswire sources:

- Agence France-Presse (*GW-afp_fr*) from May 1994 to July 2006.
- Associated Press French Service (*GW-apw_fr*) from November 1994 to July 2006.

We opted for the *GW-afp_fr* subset where we applied the following normalization rules:

- XML tags and hyphens elimination
- Lowercase conversion
- Doubled punctuation marks elimination
- Apostrophes isolation
- Substitution of (?, !, ;, :, .) into a unique "**<SEG>**" label

After normalization process of *GW-afp_fr*, the amount of tokens within the dataset was 477 million, where 9% corresponded to any punctuation mark with the "**<SEG>**" label. During experimentation, 80% of the tokens was used during training and validation while 20% was used exclusively for testing.

Results

We evaluated the performance of the models described in Section 3.1.1 with general and per class measures. *Accuracy* (Equation 3.6) is a general measure that evaluates the performance of a model regardless the class. Nevertheless, given the disparity of samples between the two classes, *Accuracy* is very likely to be biased by the **<NO SEG>** class. For this reason, *Precision* (Equation 3.7), *Recall* (Equation 3.8) and *F1-score* (Equation 3.9) measures were calculated for each one of the two classes.

[2] https://catalog.ldc.upenn.edu/LDC2006T17

Model	*Accuracy*	*Precision*		*Recall*		*F1-score*	
		<NO SEG>	**<SEG>**	**<NO SEG>**	**<SEG>**	**<NO SEG>**	**<SEG>**
CNN-A_u	0.909	0.909	0.0	1.0	0.0	0.952	0.0
CNN-A	0.963	0.972	**0.853**	**0.988**	0.718	0.980	0.778
CNN-B	**0.965**	**0.975**	0.845	0.986	**0.754**	**0.981**	**0.795**
CNN-C	0.963	0.974	0.832	0.985	0.750	0.980	0.787

Table 3.1: *CNN for Sentence Boundary Detection (French Results)*

$$Accuracy = \frac{\text{\# correctly predicted}}{\text{\# samples}} \tag{3.6}$$

$$Precision = \frac{\text{\# correctly predicted}}{\text{\#predicted}} \tag{3.7}$$

$$Recall = \frac{\text{\# correctly predicted}}{\text{\# samples}} \tag{3.8}$$

$$F1\text{-}score = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{3.9}$$

Table 3.1 shows the performance of the proposed architectures. CNN-A_u refers to the untrained CNN-A model, which presents a misleading high *Accuracy* equal to 0.909. If an analysis per class is performed, it can be seen that *Precision*, *Recall* and *F1-score* over the **<SEG>** class is zero. However, the unbalanced distribution of samples where the **<SEG>** class is just about 9%, boosts the *Accuracy* with the number of well classified samples for class **<NO SEG>**. Also in terms of *Accuracy*, CNN-A and CNN-C obtain the same score, while CNN-B slightly overperforms them. Per class metrics for **<SEG>** show an interesting behaviour, a slightly higher *Precision* is achieved by CNN-A compared to CNN-B. In contrast, for *Recall*, CNN-B overperforms CNN-A and CNN-C. This means CNN-B is more exact on predicting true borders while CNN-B achieves to cover more true borders, characteristic we consider to be more important. The harmonic mean (*F1-score*) is higher on CNN-B for both **<NO SEG>** and **<SEG>** classes.

3.1.3 Multilingual Sentence Boundary Detection

In Section 3.1.1 we presented three CNN architectures for SBD from which CNN-B showed to perform better during French experimentation. Based on these results we extended the study upon a multilingual approach to cover the rest of the languages involved in the Access Multilingual Information opinionS (AMIS) project: English and Modern Standard Arabic (MSA).

We performed English experiments with a subset of the English Gigaword Fifth Edition[3] (Parker et al., 2011b). After applying the same normalization rules than in

[3]https://catalog.ldc.upenn.edu/LDC2011T07

Language	*Accuracy*	*Precision*		*Recall*		*F1-score*	
		<NO SEG>	**<SEG>**	**<NO SEG>**	**<SEG>**	**<NO SEG>**	**<SEG>**
French	0.964	0.976	0.838	0.984	0.768	0.980	0.800
English	0.957	0.969	0.856	0.983	0.762	0.976	0.805
MSA	0.908	0.928	0.782	0.964	0.638	0.945	0.700

Table 3.2: *CNN for Sentence Boundary Detection (CNN-B Multilingual Results)*

GW-afp_fr, we selected a subset of 4.4Gb composed of near 703 million tokens where 12.13% were associated to the **<SEG>** class.

Concerning MSA, we opted for the Arabic Gigaword Fifth Edition[4] (Parker et al., 2011a). We performed a simple normalization process which consisted on doubled punctuation marks and XML tags elimination, and substitution of punctuation marks into a unique "**<SEG>**" label. We noticed that the proportion of the **<SEG>** class with respect to **<NO SEG>** was almost 10 times smaller compared to the class distributions for French and English. Which means that only 1% of the tokens were associated to the **<SEG>** class, thus causing serious issues during training of the CNN architecture. For this reason we opted to apply a downsample strategy over the **<NO SEG>** class to approach a class distribution similar to French and English ($\approx$10%) and discard all those Semantic Units (SUs) with more than 12 words. The resulting dataset after the downsampling was composed of 62.5 million tokens, where 16% were associated to the **<SEG>** class.

We also extended French experiments employing both *GW-afp_fr* and *GW-apw_fr* sources. After applying the same normalization process than in Section 3.1.2, the amount of tokens within the dataset increased to almost 587.5 million tokens.

We performed experiments with the CNN-B architecture, which showed to have better performance in Section 3.1.2. For all three languages we used the 80% of the datasets to train and 20% to test CNN-B. Results are shown in Table 3.2. Focusing on the **<SEG>** class, English presents a slightly better *Precision* than French while this second has a slightly better *Recall*, producing a minimal *F1-score* difference of 0.005. MSA has the lowest scores for all metrics. Concerning the **<SEG>** class, it presents a 0.074 *Precision* drop with respect to English and 0.130 *Recall* drop with respect to French.

3.1.4 Discussion

In Section 3.1.1 we described the three CNN architectures we used to performed SBD over French, English and MSA. CNN-A is based on the best model reported by Che et al. (2016b) evaluated over an English dataset; however, details concerning variables' initial states and tuning parameters were not specified. In Table 3.1 we showed the results related to an untrained model of the CNN-A architecture (CNN-A_u) with the idea of exemplifying how unbalanced data may mislead results. During multilingual evaluation, we presented a series of results concerning CNN-B.

[4]https://catalog.ldc.upenn.edu/LDC2011T11

Model	*Accuracy*	*Precision*		*Recall*		*F1-score*	
		<NO SEG>	**<SEG>**	**<NO SEG>**	**<SEG>**	**<NO SEG>**	**<SEG>**
CNN-2A (Che et al., 2016b)	–	–	0.776	–	0.799	–	0.788

Table 3.3: *CNN for Sentence Boundary Detection (Che et al. (2016b) English Results)*

If we focus on English, it may be useful to compare results from Table 3.2 against those obtained by Che et al. (2016b) (Table 3.3). The following considerations should be taken into account: 1) different variables' initial states and tuning parameters, 2) different training and testing datasets, and 3) different word embeddings representations. CNN-B achieves a higher *F1-score*, overperforming CNN-2A (Che et al., 2016b) in terms of *Precision*. By contrast, CNN-2A (Che et al., 2016b) overperforms CNN-B in terms of *Recall*.

3.1.5 Conclusion

In this section we justified the importance of SBD on restoring part of the syntactic information from transcripts produced by state-of-the-art Automatic Speech Recognition (ASR) systems. For this, we approached SBD as a binary classification task, where a word within a context sliding window has to be associated to one of two possible classes: **<SEG>** and **<NO SEG>**. To perform classification we opted for a CNN architecture given its capacity of extracting local features using a sliding window along an input matrix and establishing relationships between adjacent elements, which in this case are components of a Fasttext vector. Experiments were performed in a multilingual approach considering French, English and MSA.

Evaluation over French and English presented similar results, while MSA showed a lower performance. The drop in performance of MSA may be caused by a direct relation between the distribution of **<SEG>** class and the size of the sliding context window. French and English datasets presented a **<SEG>** class distribution of $\approx 10\%$. By contrast, **<SEG>** class for MSA was originally $\approx 1\%$. For all three languages we set the size of the sliding context window to 5 words, which was enough to capture the context around the SU boundaries for French and English but small to MSA.

The best CNN architecture and trained models for all three languages have been included as part of the SBD module in the context of the AMIS project (Smaïli et al., 2018; Grega et al., 2019; Smaïli et al., 2019).

3.2 Sentence Boundary Detection for Modern Standard Arabic Transcripts

Arabic language is known to be challenging given its complex linguistic structure (Attia and Somers, 2008) and dialect variations (Habash, 2010). However, the development of Arabic Natural Language Processing (NLP) tools has increased these last years, creating a large set of state-of-the-art applications including Part-of-Speech (POS) taggers, syntactic parsers, Machine Translation (MT), Automatic Speech Recognition (ASR) and automatic speech synthesis systems (Farghaly and Shaalan, 2009; Jaafar and Bouzoubaa, 2018). Some NLP libraries and tools like Python NLTK[5], OpenNLP[6] (Baldridge, 2005), UIMA[7] (Ferrucci and Lally, 2004), LIMA[8] and NooJ[9] (Silberztein, 2005), which were originally created for non Arabic texts now include Arabic extensions. By contrast, other libraries have been developed exclusively for Arabic. In the following paragraphs we present an overview of some available NLP tools for Arabic.

Althobaiti et al. (2014) developed AraNLP, which is focused on Arabic preprocessing. This library contains some tools covering tokenization, stemming, POS tagging, sentence detection, word segmentation, normalization and punctuation/diacritic deletion.

MADAMIRA[10], developed by Pasha et al. (2014), is a toolkit which combines two previous Arabic NLP systems: MADA (Habash et al., 2009) and AMIRA (Diab, 2009). It provides the following NLP tools for Modern Standard Arabic (MSA) and the Egyptian Arabic dialect: lemmatization, diacritization, glossing, POS tagging, morphological analysis, morphological disambiguation, stemming, tokenization, base phrase chunking, name entity recognition and word-level disambiguation.

SAFAR (Souteh and Bouzoubaa, 2011) is a framework that aims to gather developed Arabic NLP tools within a single homogeneous architecture and create new ones if necessary. Implemented tools within SAFAR include morphological analyzers, stemmers, syntactic parsers, normalizers, tokenizers, sentence splitters, transliteration tools and question answering applications.

Research has also been done regarding unstructured and noisy Arabic texts found in social media datasets like microblogs and tweets. Added to the common difficulties of working with structured Arabic texts, Arabic tweets carry other problems like high degree of ambiguity and spelling mistakes. Mallek et al. (2018) implemented a phrase-based statistical MT system from MSA tweets into English. They concluded that a preprocessing step for this kind of noisy texts is very useful to improve the translation results. El-Masri et al. (2017) presented a sentiment analysis tool over Arabic tweets

[5] https://www.nltk.org/
[6] https://opennlp.apache.org/
[7] https://uima.apache.org/
[8] https://github.com/aymara/lima/wiki
[9] http://www.nooj-association.org/
[10] https://camel.abudhabi.nyu.edu/madamira/

using a Naive Bayes learning approach. Their model detects the polarity of a group of tweet classifying it in four possible classes: positive, negative, both and neutral.

Access to nowadays available Internet multimedia platforms like Youtube[11], TED[12] and Dailymotion[13] has opened a new universe for Arabic NLP tools. ASR systems can be used to transcribe multimedia content, thus enabling natural human-machine interaction with further NLP tasks (Yu and Deng, 2016). Performance of ASR systems for MSA has improve in the last years given the amount of data available for training and testing these systems. Tomashenko et al. (2016) used features derived from Gaussian Mixture Models to train a Neural Network (NN) combined with the use of time-delay NNs for acoustic modeling over 1 128 hours of MSA broadcast speech and 110 million words, reporting a Word Error Rate (WER) of 23%. Menacer et al. (2017b) developed an ASR system for MSA based on the Kaldi toolkit[14] (Povey et al., 2011), where recognition is achieved using a NN + Hidden Markov Model (HMM) over 63 hours of spoken transcribed data and 1 000 million words from the Arabic Gigaword corpus. They reported a result of 14.42% in terms of WER. Despite the good performance of modern ASR systems for Arabic, transcripts do not carry syntactic information as sentence boundaries, which is a major problem for further NLP tasks.

Sentence Boundary Detection (SBD) is of vital importance given that in general, ASR systems focus on obtaining the correct sequence of transcribed words with almost no concern of the overall structure of the document, thus lacking of syntactic information (Gotoh and Renals, 2000). A big complication of segmenting a transcript is the flurry definition of sentence in spoken language. In standard conversations, even in simpler scenarios like a monologue, ideas are organized very different compared to written language. Added to this, Arabic carries other difficulties like word ambiguity, structural ambiguity, lack of punctuation marks, use of connective words and agglutination (Hadrich et al., 2005).

To our knowledge not much work has been developed over Arabic SBD. The Arabic Texts Segmentation System or STAr (by its acronym in French), created by Hadrich et al. (2005), is a text segmentation system for Arabic based on a set of rules created from the contextual analysis of punctuation marks and a list of particles which play the role of sentence boundaries. For STAr, segmentation consists in the disambiguation of sentence boundaries and paragraphs. Even though they did not report any experiment with transcripts, it is possible to apply the method over this type of spoken texts.

Zribi et al. (2016) presented three methods for the detection of sentence boundaries in transcribed Tunisian Arabic using lexical and prosodic features. The first method is composed of two sets of handmade rules: 1) based on oral specific lexical items and prosodic features; and 2) based on connectors, personal and relative pronouns, verbs, etc. The second method corresponds to a statistical method based on a decision tree algorithm. It classifies a word into four different classes: 1) first word of a sentence, 2)

[11] `https://www.youtube.com/`
[12] `https://www.ted.com/`
[13] `https://www.dailymotion.com/fr`
[14] `http://kaldi-asr.org/`

word within a sentence, 3) last word of a sentence and 4) one word sentence. The third method combines the previous two in a hybrid framework.

3.2.1 Experimental Evaluation

We implemented two different SBD systems based on NNs. The first system corresponds to CNN-B, described in Section 3.1.1. The second system, based on Tran et al. (2016); Linhares Pontes et al. (2018), is a Long Short Term Memory (LSTM) that follows a sequence-to-sequence paradigm using the attention mechanism to verify which words of a sequence represent a Semantic Unit (SU) boundary (González-Gallardo et al., 2018b).

Dataset

One of the objectives of this chapter is to analyze the impact of cross-domain datasets during the evaluation phase of SBD. The first dataset (*GW-aaw_ar*) corresponds to the Asharq Al-Awsat (aaw_arb) news wire of the Arabic Gigaword Fifth Edition (Parker et al., 2011a), which after XML extraction is composed of $\approx$ 73 million words. The second dataset (*TED_ar*) corresponds to the dataset from the Multilingual Task 2017 proposed by The International Workshop on Spoken Language Translation (IWSLT)[15]. It consists of 122 manually transcribed TED talks[16] in MSA containing $\approx$ 168 thousand words. During the normalization phase, all punctuation marks (؟ ? ! ! ، , . . : ;) were mapped to a common boundary symbol, which corresponds to the **<SEG>** class. Based on the experiments performed by Alotaiby et al. (2010), where they observed that a big lexicon could be reduced about 24.54%, we used the MADAMIRA toolkit to perform a tokenization over both datasets to reduce their dimensionality. The following proclitics and enclitics were separated, generating two or more tokens depending of the amount of clitics within the word:

ال ، و ، ف ، ل ، ب ، ك ، س ، ي ، ا ، ني ، كمَا ، كم ، كن ، هَ ، همَا ، هن ، هم ، نَا

Table 3.4 shows the final number of tokens for both datasets after normalization and tokenization, as well as the training, validation and testing distributions. We opted to omit the validation set for *TED_ar* given its reduced size. Class distribution is different for both datasets. For *Gw-aaw_ar*, 6% of samples correspond to the **<SEG>** class; while 10% for *TED_ar*.

We conducted two experimental scenarios for SBD over MSA. Similar to previous experiments, we opted for Fasttext vectors to represent our datasets and conduct our experiments given its advantages concerning morphology rich languages. We performed a 300 dimension vector induction with the complete *Gw-aaw_ar* dataset obtain-

[15] http://workshop2017.iwslt.org/

[16] https://www.ted.com/talks?language=ar

Dataset	Train	Valid	Test	Total
Gw-aaw_ar	73 608 328	21 030 957	10 515 477	105 154 762
TED_ar	183 314	-	50 881	234 195

Table 3.4: *Size and Distribution of Datasets*

Dataset	Model	*Accuracy*	*Precision*		*Recall*		*F1-score*	
			<NO SEG>	**<SEG>**	**<NO SEG>**	**<SEG>**	**<NO SEG>**	**<SEG>**
GW-aaw_ar$_{test}$	CNN-B	0.963	0.972	0.797	0.989	0.612	0.980	0.684
	LSTM	0.947	0.954	0.729	0.991	0.327	0.972	0.451
TED_ar$_{test}$	CNN-B	0.934	0.945	0.752	0.983	0.471	0.964	0.579
	LSTM	0.914	0.921	0.673	0.989	0.211	0.954	0.321

Table 3.5: *Ex.1 Results*

ing 102 248 vectors. Vectors of out-of-vocabulary (OOV) words from the embedding model are generated from the word's *n*-grams vectors, eliminating unknown vectors.

Experiment 1 (EX.1)

With this first experiment we wanted to observe the impact of applying a SBD model trained with a big dataset collected from written sources over a spoken source dataset. We first conducted training, validation and test on both CNN-B and LSTM systems with *GW-aaw_ar*$_{train,valid,test}$ for 3 and 7 epochs respectively. The number of epochs were dynamically decided with *GW-aaw_ar*$_{valid}$ before overfitting. Then, we used *TED_ar*$_{test}$ for evaluating the performance of the trained models over an out-of-domain dataset.

Results for this scenario are shown in Table 3.5. As discussed in previous experiments, high *Accuracy* results may give an erroneous idea of the model performance. Both CNN-B and LSTM perform really good concerning the **<NO SEG>** class over *GW-aar_ar*$_{test}$. For **<SEG>** class, *Precision* of CNN-B is slightly higher than LSTM; but the biggest difference concerns *Recall* where CNN-B performs almost two times better. Evaluation of the models over *TED_ar*$_{test}$ shows a interesting behaviour when compared to *GW-aar_ar*$_{test}$. It is possible to appreciate a small decrease in *Precision* for both models; nevertheless, *Recall* drops 0.141 for CNN-B and 0.116 for LSTM.

Experiment 2 (EX.2)

The objective of the second experiment was to measure the effect of adding a small in-domain spoken dataset over the models trained on EX.1. For this experiment we continued training CNN-B and LSTM systems with *TED_ar*$_{train}$. *TED_ar* dataset size is very small for NN training strategies to consider the creation of a validation set. For this reason, *GW-aaw_ar*$_{valid}$ was used during validation phase and epoch control for both systems. The reduced size of *TED*$_{train}$ lead to a fast overfitting behaviour; therefore, both CNN-B and LSTM were trained only for one epoch. Evaluation was performed over the same dataset of EX.1 (*TED_ar*$_{test}$).

Model	*Accuracy*	*Precision*		*Recall*		*F1-score*	
		<NO SEG>	**<SEG>**	**<NO SEG>**	**<SEG>**	**<NO SEG>**	**<SEG>**
CNN-B	0.938	0.963	0.687	0.968	0.655	0.966	0.671
LSTM	0.911	0.925	0.597	0.981	0.264	0.952	0.366

Table 3.6: *Ex.2 Results*

Table 3.6 shows the results for EX.2. Similar to EX.1, *Accuracy* values are very high given class unbalanced. Concerning CNN-B, *Precision* and *Recall* for the **<NO SEG>** class are almost the same. Continue training SBD_{Conv} with TED_{train} seems to have a negative impact over the **<SEG>** class *Precision*, which is lower than in EX.1; however, *Recall* improves. LSTM shows a similar behavior than CNN-B. A slight improvement of *Recall* for the **<SEG>** class is present, yet a decrease for *Precision* is produced.

3.2.2 Conclusion

In this section we studied the impact of using cross-domain datasets during the evaluation phase of two SBD systems over MSA. The obtained results show that tuning a model that was originally trained with a big out-of-domain dataset with small in-domain dataset, in general, improves its performance.

Results for EX.1 and EX.2 reflect how unbalanced classes distribution impact SBD systems in a similar degree even both methods follow different learning techniques. CNN-B focus its attention on analyzing the words contained in a fixed-sized window, making the boundary prediction independent of the actual position within the transcript. Nevertheless, this advantage is also a drawback for potentially long sentences given that the method is not able to analyze long contexts. By contrast, LSTM is characterized by the analysis of a sequence of words to propose the sentence boundary of this sequence. This approach works best when it analyzes a sequence of words at the beginning of sentences. However, LSTM analyzes word sequences that can start in the middle or at the end of sentences, which reduces the performance of predicting sentence boundaries. In addition, long and complex sentences are a challenge to code all the information and to generate a correct sentence boundary for this kind of sentences.

LSTM showed to be less effective compared to CNN-B during the evaluation of *GW-aaw_ar*; nevertheless, both systems presented a similar drop in performance when TED_{ar} was evaluated. After adding the in-domain dataset into both systems, CNN-B exhibited an important improvement in terms of *F1-score*. Regarding LSTM, improvement was not very big; which may be caused by the small size of the in-domain training dataset. We can conclude that adding a small in-domain dataset to a model trained with a larger out-of-domain dataset improves the performance of the system; still, each system will improve in different ratios depending of its learning strategy.

3.3 Sentence Boundary Detection with Transcription Errors

Automatic Speech Recognition (ASR) systems produce good quality transcripts which nowadays are just the first step of Natural Language Processing (NLP) pipelines. An example is Cross-Lingual Speech-to-Text Summarization (Pontes et al., 2018), where a transcript in a source language is segmented by a Sentence Boundary Detection (SBD) system; then a summarization process produces an informative and condensed version of the segmented transcript which is finally translated into a target language by a Machine Translation (MT) system. While this kind of pipelines are very useful and provide people with valuable information that in other circumstances access would be impossible, it is challenging given the complications each element of the pipeline faces. In this section we focus in the first part of the pipeline and study how SBD deal with transcription errors in a controlled environment.

3.3.1 Dataset

The MultiLing Pilot 2011 dataset is a collection of WikiNews texts originally in English that were translated into Arabic, Czech, English, French, Greek, Hebrew and Hindi by native speakers (Giannakopoulos et al., 2011). Each language version of this dataset is composed of 10 topics where each topic is consists of 10 source texts and 3 reference summaries with a maximum of 250 words. The experiments we perform in this section are driven with the French version of this dataset ($MultiLing_{fr}$).

3.3.2 Experimental Results

The quality of transcripts produced by ASR systems is normally measured in terms of Word Error Rate (WER) by comparing the resulting transcript against one or more references. This measure considers three different errors and calculates a general value indicating the quality of the transcript; the lower the value (closer to zero), the higher its quality. The three errors considered by WER are deletions, insertions, and substitutions. WER is defined as:

$$WER = \frac{D + I + S}{N} \tag{3.10}$$

where D corresponds to the number of deletions, I to the number of insertions, S to the number of substitutions and N to the number of words in the reference. An automatic transcript carries all three errors at different ratios; yet, for this controlled scenario we simulated in an isolated way each error to observe how each of them affects the performance of the SBD process.

We approximated WER by simulating the errors produced by ASR systems in a straightforward approach. We created the deletion error dataset ($MultiLing\text{-}D_{fr}$) by

Dataset	Class	*Precision*	*Recall*	*F1-score*
MultiLing$_{fr}$	**<NO SEG>**	0.971	0.986	0.978
	<SEG>	0.840	0.721	0.776
MultiLing-D$_{fr}$	**<NO SEG>**	0.966	0.963	0.965
	<SEG>	0.654	0.673	0.663
MultiLing-I$_{fr}$	**<NO SEG>**	0.960	0.956	0.958
	<SEG>	0.592	0.616	0.604
MultiLing-S$_{fr}$	**<NO SEG>**	0.958	0.950	0.954
	<SEG>	0.554	0.600	0.576

Table 3.7: *Results of Sentence Boundary Detection with Transcription Errors*

choosing randomly and deleting m words of each document in *MultiLing-D*$_{fr}$. Concerning the substitution error dataset (*MultiLing-S*$_{fr}$), for each document we first selected a set $Y = \{y_1, y_2, \ldots, y_m\}$ of words randomly. Then, for each word w_i of the document, a randomly generated decision value $v_i \in [0, 1]$ was calculated. Finally, if v_i happened to be greater than a threshold equal to 0.5, w_i was replaced by y_j. This cycle was repeated until all words y_j in Y where picked. With regards to the insertion error dataset (*MultiLing-I*$_{fr}$), we followed the same procedure as for *MultiLing-D*$_{fr}$; but instead of replacing w_i by y_j,we placed y_j after w_i. For all three error datasets, m was calculated as:

$$m = WER \times N \tag{3.11}$$

where N corresponds to the length (number of words) in each original document and WER was fixed to 0.15.

We simulated the lack of punctuation by deleting all punctuation signs inside *MultiLing*$_{fr}$ and the datasets with induced transcription errors (*MultiLing-D*$_{fr}$, *MultiLing-S*$_{fr}$, *MultiLing-I*$_{fr}$). Then, we automatically restored them with the CNN-B architecture presented in Section 3.1.1, following the same dataset and preprocessing steps than in Section 3.1.3.

Table 3.7 presents in terms of *Precision*, *Recall* and *F1-score* the automatic evaluation performed over the unpunctuated datasets. As seen from the **<NO SEG>** class, the method has a really good performance (over 0.95 for all metrics) no matter the type of transcription error; which is an expected behaviour given the unbalanced nature of the data. Nevertheless, for the **<SEG>** class the performance drops when trying to segment the noisy transcripts. The worst scenario corresponds to the dataset with substitution errors (*MultiLing-S*$_{fr}$), where *Precision* and *Recall* present relative drops of 34% and 17% with respect to *MultiLing*$_{fr}$.

3.3.3 Conclusion

In this section we studied the impact that transcription errors have over SBD. We created a controlled experimental environment where we simulated each one of the possible errors an automatic transcript may produce: deletions, insertions and substitutions. For each error we fixed a $WER = 0.15$, which is an acceptable value for standard ASR systems. Results showed that all three types of errors had a negative impact over CNN-B performance.

In general terms, for each prediction of CNN-B, all words inside the sliding window get merged by the convolution and pooling layers by creating a set of general features. In the case of deletion errors, if one of the words is missing, the rest of the context may recover from that loss in a certain degree. This recovery is more difficult for insertion errors, where one or more words get inside the sliding window, causing a highly improbable context. Finally, substitution errors have a similar behavior that insertion errors with the difference that in this case the original word completely disappears from the sliding window. Substitution errors can be seen as deletion with insertion errors occurring at the same time.

Chapter 4

Text-based Multimedia Summarization for Modern Standard Arabic

Contents

Arabic is the official language of 22 countries and is now the fifth spoken language with more than 300 million speakers (Azmi and Altmami, 2018; Qaroush et al., 2019). It is also the fastest growing language in the Web during the last 19 years in number of Internet users[1] (Miniwatts Marketting Group, 2019). This factors have lead to a massive amount of online multimedia documents which motivates the necessity of developing tools that help processing all this information. Natural Language Processing (NLP) for Arabic is more challenging than other languages like English and French given its complexity in terms of morphology and structure. The following list synthesizes the five main complications that are present in Arabic and how they difficult the application of NLP tasks according to Al-Saleh and Menai (2016):

- Given that Arabic is highly inflectional and derivational, morphological analysis such as lemmatization and stemming become really complex.

[1] https://www.internetworldstats.com/stats7.htm

- Capitalization is nonexistent, making challenging the process of name entities recognition.
- It is common that Arabic texts omit diacritics, increasing the complexity of inferring meaning.
- Arabic is highly ambiguous in comparison to other languages.
- There is not a lot of public corpora available, restricting training and complicating evaluation of NLP tasks.

In Section 1.2, we explained the four architectures that are proposed by the Access Multilingual Information opinionS (AMIS) project to summarize a video whether in French or Modern Standard Arabic (MSA) into English. Two of these architectures (SC3 & SC4) are focused on a Text-based multimedia summarization approach, establishing the need of developing an Automatic Text Summarization (ATS) system capable of producing MSA extractive summaries.

This chapter is divided in two main sections. In Section 4.1 we first introduce ARTEX for Modern Standard Arabic (ARTEX-MSA), an extension to Autre Résumeur de TEXtes (ARTEX) capable of generating extractive summaries in MSA. Then, we explain the EASC dataset, which we used to evaluate and compare ARTEX-MSA. Finally, we present some comparative results of summaries with different Compression Ratios (CRs) based on Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores. Then, in Section 4.2 we conduct a study to evaluate the performance of ARTEX-MSA over MSA automatic transcripts.

4.1 ARTEX for Modern Standard Arabic (ARTEX-MSA)

Autre Résumeur de TEXtes (ARTEX) is an extractive algorithm for Automatic Text Summarization (ATS) developed by Torres-Moreno (2012a). Conceived as a language independent summarizer, ARTEX is currently able to summarize documents in English, French and Spanish. ARTEX is divided in two main phases. A preprocessing phase, in charge of representing the document into a suitable space and a summarizing phase, which performs the summarization methodology.

Document Preprocessing

ARTEX follows an extractive summarization approach, thus the source document has to be segmented into cohesive textual segments that will be then assembled to produce the summary. During this phase, ARTEX performs the four following steps.

1. **Sentence Splitting**: The source document is divided into sentences based on punctuation marks.

2. **Sentence Filtering**: Words with less than two occurrences, functional words and very common words are removed using external resources like stops-lists and dictionaries.

3. **Word Normalization**: Words are normalized to their canonical form using lemmatization, stemming or ultra-stemming. Lemmatization is performed with language-dependent dictionaries of morphological families. Stemming is performed with the Porter Stemmer algorithm (Van Rijsbergen et al., 1980). Ultra-steming (Torres-Moreno, 2012b) considers only the n first letters of a word.

4. **Text Vectorization**: The source document is represented as a Vector Space Model (VSM) with a $S^{[P\times N]}$ matrix of P sentences and N characteristics. Each component $s_{\mu,j}$ represents the occurrences of the characteristic (word, lemma, stem or ultra-stem) j in the sentence s_μ.

Summarization methodology

The matrix $S^{[P\times N]}$ is used during this phase by ARTEX to compute the score ω of each sentence s_μ by calculating the inner product between a sentence vector $\mathbf{s}_\mu$, an average pseudo-sentence vector **b** (global topic) and an average pseudo-word vector **a** (lexical weight). The global topic and the lexical weight vectors are shown in Figure 4.1. Concerning the global topic, it is represented in a N dimensional space of words (Figure 4.1 (A)); while the lexical weight vector is represented in a VSM of P sentences (Figure 4.1 (B)).

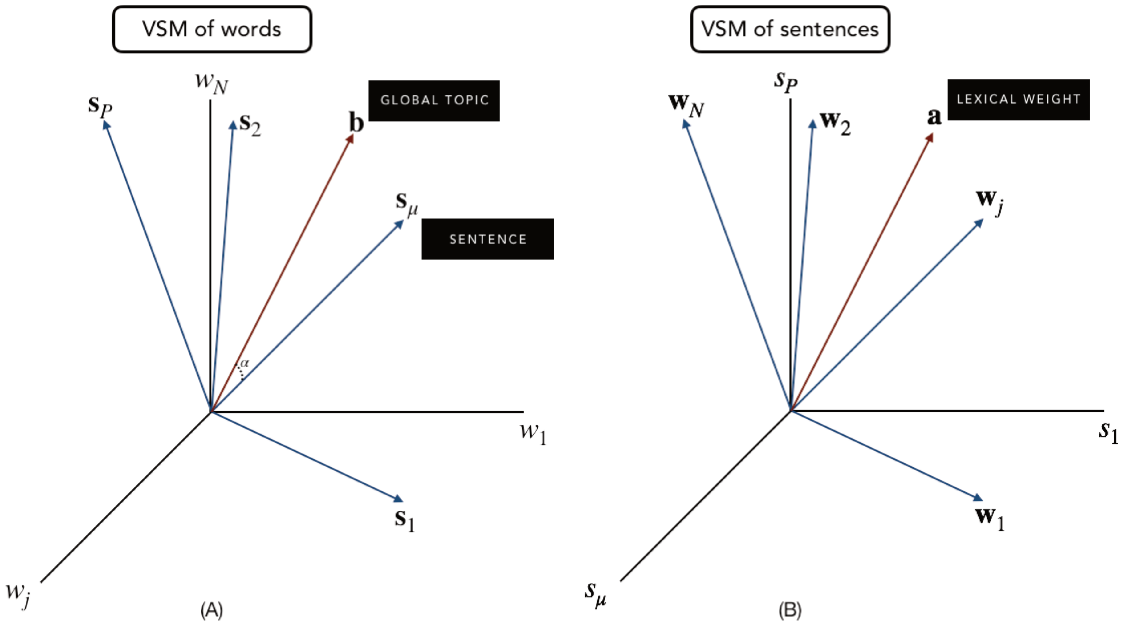

Figure 4.1: *Global Topic and Lexical Weight Vectors for ARTEX*

Let $\mathbf{s}_\mu = \langle s_1, s_2, \ldots, s_N \rangle$ be a vector of the sentence s_μ for $\mu = 1, 2, \ldots, P$. The average pseudo-word vector $\mathbf{a} = \langle a_1, a_2, \ldots, a_P \rangle$ is defined as the average number of occurrences of N words used in sentence μ:

$$a_\mu = \frac{1}{N} \sum_{j}^{N} s_{\mu,j} \quad ; \quad \mu = 1, 2, \ldots, P \tag{4.1}$$

and the average pseudo-sentence vector $\mathbf{b} = \langle b_1, b_2, \ldots, b_N \rangle$ as the average number of occurrences of each word j used trough the P sentences:

$$b_j = \frac{1}{P} \sum_{j}^{P} s_{\mu,j} \quad ; \quad j = 1, 2, \ldots, N \tag{4.2}$$

The score ω for a sentence s_μ is formally defined as follows:

$$\omega(s_\mu) = (\mathbf{s}_\mu \cdot \mathbf{b}) \cdot \mathbf{a} = \frac{1}{NP} \sum_{j}^{N} \left(s_{\mu,j} \cdot b_j \right) \cdot a_\mu \tag{4.3}$$

After computing ω for all sentences s_μ, the summary is produced by concatenating the sentences with the highest scores in their original order of appearance until the desired Compression Ratio (CR) is reached.

We conceived ARTEX for Modern Standard Arabic (ARTEX-MSA) by extending ARTEX functionality to process Modern Standard Arabic (MSA). ARTEX has the great advantage of being lightweight and easy to extend to other languages. In fact, their only language dependent components are "Sentence Filtering" and "World Normalization" during the document preprocessing phase. ARTEX-MSA focuses on this two steps.

With regards to "Sentence Filtering", we opted for a public available Arabic stop-list compilation [2] composed of 750 words. This words carry little meaning and serve only a syntactic function. A stop-list of 750 words may seem big compared to other languages like English, French or Spanish, whose stop-lists contain no more than 300 entries. However, given that Arabic is a language with rich morphology, one stop-word may have different morphological variants. Other stop-lists have been reported in literature (El-Khair, 2006; Azmi and Al-Thanyyan, 2012; Althobaiti et al., 2014); nonetheless, they are not public available or contain a reduce about of words.

[2] `https://github.com/mohataher/arabic-stop-words`

Arabic morphology consists primarily of a system of consonant roots which interlock with patterns of vowels to form words or word stems. Ryding (2005) defined a root as:

"... a relatively invariable discontinuous bound morpheme, represented by two to five phonemes, typically three consonants in a certain order, which interlocks with a pattern to form a stem and which has lexical meaning."

and a pattern as:

"... a bound and in many cases, discontinuous morpheme consisting of one or more vowels and slots for root phonemes (radicals), which either alone or in combination with one to three derivational affixes, interlocks with a root to form a stem, and which generally has grammatical meaning."

Based on this two definitions, Arabic word normalization consists on removing the patters present in a word to leave only a stem composed of three, four or five consonants. Concerning the "Word Normalization" step, we included and adapted a MSA stemming module[3] which performs the two following actions:

1. Prefixes and suffixes are discarded, leaving only the root of the word. Suffixes relative to people like possessive adjectives are not discarded.
2. Unnecessary letters are stripped-down depending of the length of the word. This action follows an iterative process always trying to obtain the shortest possible valid stem for the corresponding word.

After the "Word Normalization" step, the rest of ARTEX-MSA's summarization methodology is performed in the same way as with standard ARTEX.

4.1.1 Dataset

Automatic evaluation of MSA summarization is challenging given the lack of existing MSA evaluation datasets (Qaroush et al., 2019). One of the few available dataset is Essex Arabic Summaries Corpus (EASC), created by El-Haj et al. (2010). This dataset comprises a set of 153 documents extracted from the Arabic Wikipedia[4], and the newspapers Alrai[5] (Jordan) and Alwatan6[6] (Saudi Arabia). Each document contains in average 380 words.

For each document five annotators were asked to read and create an extractive summary by selecting at most half of the sentences within the document. Similar to previous research (Qaroush et al., 2019), we opted to create a summary gold-standard dataset (*EASC-Gold*) through a voting process among all five references. For each set of five

[3] https://github.com/arnoo/arstem
[4] https://ar.wikipedia.org
[5] http://alrai.com/
[6] https://www.alwatan.com.sa/

Dataset	CR	ROUGE-1	ROUGE-2
EASC-Gold	25%	0.474	0.331
	30%	0.501	0.358
	35%	0.532	0.388
	50%	0.581	0.441
EASC-Gold_set30	25%	0.458	0.306
	30%	0.483	0.330
	35%	0.521	0.370
	50%	0.558	0.400
EASC-Gold30	25%	0.523	0.414
	30%	0.553	0.438
	35%	0.581	0.466
	50%	0.651	0.539

Table 4.1: *ROUGE-1 and ROUGE-2 Evaluation Over EASC-Gold and EASC-Gold30*

summaries that correspond to the same document, the gold-standard summary is composed by those sentences that exist in at least three summaries. Given that annotators had the freedom to create summaries with CRs $\leq$ 50%, the CR of the resulting summaries from *EASC-Gold* is variable. Nevertheless, in average, the CR for *EASC-Gold* is 28.1%.

After a manual analysis of *EASC-Gold* we found that all diacritics had been eliminated and a big amount of summaries had incomplete sentences; this second observation is relevant because it may produce incoherences between reference and automatically produced summaries leading to misleading evaluation results. For this reason we decided to create a sampled gold-standard (*EASC-Gold30*) by picking a random sample of 30 documents with their corresponding gold-standard summaries and manually restoring diacritics and completing those incomplete sentences. The CR of this random sample is 28.3%, similar to the one of the complete gold-standard dataset.

4.1.2 Experimental Evaluation

Given that the EASC dataset does not have a defined CR ($\leq$ 50%), we opted to use ARTEX-MSA to summarize the source documents from *EASC-Gold* and *EASC-Gold30* with CRs between 25% and 50%. To analyze the impact that incomplete sentences from *EASC-Gold*'s reference summaries have over automatic evaluation, we performed different automatic evaluations with ROUGE-1 and ROUGE-2. We first evaluated the summaries generated by ARTEX-MSA from *EASC-Gold* against the corresponding *EASC-Gold*'s gold-standard summaries (top section of Table 4.1). We then evaluated the summaries generated by ARTEX-MSA but only from the corresponding 30 samples of *EASC-Gold30*; always with the *EASC-Gold* dataset (middle section of Table 4.1). Finally we evaluated the summaries generated by ARTEX-MSA from *EASC-Gold30* against the corresponding *EASC-Gold30*'s gold-standard summaries (bottom section of Table 4.1).

System	ROUGE-2
(Al-Radaideh and Afif, 2009)	0.161
(Haboush et al., 2012)	0.180
LCEAS (Al-Khawaldeh and Samawi, 2015)	0.271
mRMR (Oufaida et al., 2014)	0.282
AQBTSS (El-Haj et al., 2010)	0.445
(Al-Abdallah and Al-Taani, 2017)	0.449
(Al-Radaideh and Bataineh, 2018)	0.465
ARTEX-MSA	0.531
ESMAT (Binwahlan, 2015)	0.589
Gen-Summ (El-Haj et al., 2010)	0.599
LSA-Summ (El-Haj et al., 2010)	0.605
Score-based (Qaroush et al., 2019)	0.633
ML-based (Qaroush et al., 2019)	0.783

Table 4.2: *Performance Comparison Over EASC-Gold*

Results between *EASC-Gold* and *EASC-Gold_set30* shown in Table 4.1 (top and middle sections) present a small difference for both ROUGE-1 and ROUGE-2; however, this difference is not statistically significant ($p > 0.05$). This is a good indicator that the 30 videos chosen from *EASC-Gold* to create *EASC-Gold30* were uniformly distributed over the dataset and represent correctly the complete *EASC-Gold* dataset. If we focus on the bottom section of Table 4.1, that corresponds to the evaluation of the summaries generated by ARTEX-MSA from *EASC-Gold30* against the corresponding *EASC-Gold30*'s gold-standard summaries, we observe a statistically significant ($p < 0.01$) improvement compared to *EASC-Gold* and *EASC-Gold_set30*.

A comparison of ARTEX-MSA and different summarization systems using ROUGE-2 over the *EASC-Gold* dataset is shown in Table 4.2. Setting a CR is not possible given that the CRs from *EASC-Gold* summaries are not fixed; for this reason, summaries produced by the summarization systems had the only restriction of a CR $\leq$ 50%. In the case of ARTEX-MSA, for each summary in *EASC-Gold30*, we opted to take into account the CR (25%, 30%, 35% or 50%) that obtained the best ROUGE-2 to compute the mean values.

Discussion

Compared to the rest of the systems in Table 4.2, ARTEX-MSA shows an average performance; however, some considerations should be taken into account. AQBTSS (El-Haj et al., 2010), LSA-Summ (El-Haj et al., 2010), Gen-Summ (El-Haj et al., 2010) and ESMAT (Binwahlan, 2015) systems were evaluated based on ROUGE-cut 100. A ROUGE modification that takes into account only the first 100 words from the beginning of each summary. Concerning ML-Based (Qaroush et al., 2019) and Score-based (Qaroush et al., 2019), each summary was set to match the same number of words than the corresponding *EASC-Gold* summary. With respect to ARTEX-MSA, for each summary we considered the CR (25%, 30%, 35%, 50%) that obtained the best ROUGE-2 score. Yet

this three strategies have an effect over performance in different ratios, we think this ranking provides an adequate overview.

4.1.3 Conclusion

ATS for MSA is far developed from other languages like English or French given its complexity in terms of morphology and structure. In this Section we presented ARTEX-MSA, an extractive ATS system for MSA by adapting the preprocessing phase of ARTEX. Comparative results over the EASC dataset showed that ARTEX-MSA has an average performance. However, it has the advantage of being lightweight, portable and language scalable.

ARTEX-MSA has been integrated into the Access Multilingual Information opinionS (AMIS) project as part of architectures SC3 and SC4 (Smaïli et al., 2018; Grega et al., 2019; Smaïli et al., 2019).

4.2 Extractive Text-based Multimedia Summarization for Modern Standard Arabic

In Section 1.2, we explained that one of the components of the Access Multilingual Information opinionS (AMIS) project consists of a text-based summarizer capable of processing English, French and Arabic Automatic Speech Recognition (ASR) transcripts. In this section we apply ARTEX for Modern Standard Arabic (ARTEX-MSA) over a set of samples from the *AMIS-Dataset* to study how different Compression Ratios (CRs) affect selected segments distribution. We also perform an automatic evaluation of ARTEX-MSA, which we compared with a baseline using FRamework for Evaluating Summaries Automatically (FRESA).

Figure 4.2 shows the full pipeline of extractive text-based multimedia summarization for Modern Standard Arabic (MSA). An ASR system first performs a raw transcript from the audio document. Then, a Sentence Boundary Detection (SBD) system is in charge of segmenting the transcript into Semantic Units (SUs). Finally, ARTEX-MSA preprocess the segmented transcript and performs summarization based on the desired CR. It is important to mention the potential domino effect that errors in an early step could impact the performance of the resulting summary.

4.2.1 Dataset

We gathered a set of 30 samples in Arabic (*AMIS-Dataset30*) from the *AMIS-Dataset*. We first manually analyzed all selected samples to assure that MSA was the only Arabic variation present in the videos. We then applied an ASR process over the selected videos with the Arabic Loria Automatic Speech Recognition (ALASR) system, developed by Menacer et al. (2017a) at LORIA Laboratory as part of the AMIS project.

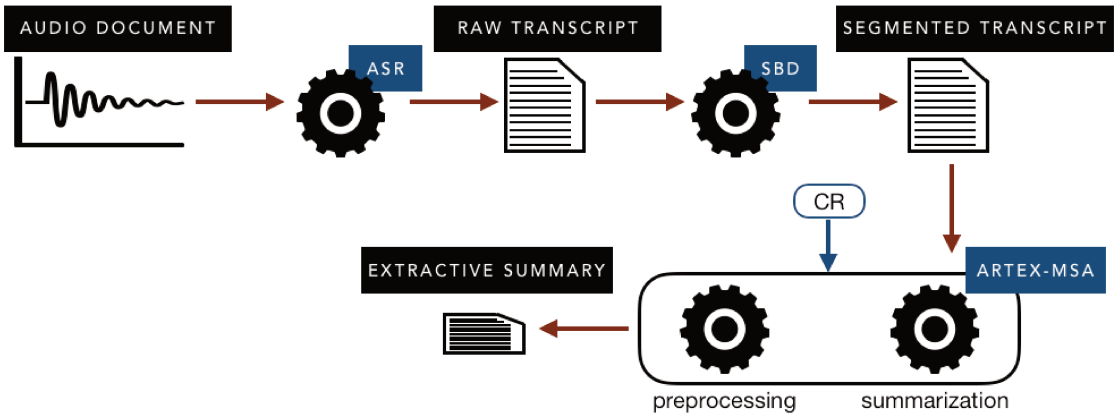

Figure 4.2: *Extractive Text-based Multimedia Summarization for MSA*

ALASR presents similar performance than state-of-the-art systems (Tomashenko et al., 2016; AlHanai et al., 2016; Alsharhan and Ramsay, 2017) with a Word Error Rate (WER) equal to 14.02%.

In average, the *AMIS-Dataset30* is composed of 438 words per sample, where the shortest one contains only 52 words, and the longest one 3 695 words. After transcription process we applied the SBD system presented in Section 3.1.1 (CNN-B) with the model trained in Section 3.2.1. In average, transcripts are constituted of 43 SUs with 13 words each. Figures A.1 and A.2 show the number of SUs and the average SU length per sample.

4.2.2 Experimental Evaluation

We generated a total of 90 summaries with ARTEX-MSA over *AMIS-Dataset30*. To compare how does the Automatic Text Summarization (ATS) system behaved depending of the size of the summaries to be created, we defined three different CRs: 25%, 30% and 35%. Table 4.3 shows some statistics related to the different CRs that were established to generate the summaries. While the average number of segments increases with bigger CRs, the average number of words per segment decreases. We were also interested in analyzing for each CR the distribution of the selected segments over their corresponding transcript. For this, we divided each transcript into beginning, middle, and end; each division corresponding to one third of the complete document and counted the number of segments belonging to each division. Segments distribution can be seen in the last three columns of Table 4.3. ARTEX-MSA seemed to select segments over the transcript in equal ratios; yet there is a small difference between divisions, this difference is not statistically significant ($p > 0.05$).

We created baseline summaries by selecting the first n lines from the transcripts depending of the desired CR to measure and compare the performance of ARTEX-MSA over *AMIS-Dataset30*. We evaluated the baseline system and ARTEX-MSA by comparing the divergence between the probability distributions of each summary against its

CR	Average Number of Segments	Average Segments Length	Segments Distribution		
			Beginning	Middle	End
25%	12.367	18.340	0.342	0.356	0.302
30%	14.800	17.179	0.336	0.356	0.309
35%	16.933	16.627	0.341	0.344	0.315

Table 4.3: *Summary Statistics of ARTEX-MSA over AMIS-Dataset30*

System	CR	FRESA 1	FRESA 2	FRESA 4	FRESA N
Baseline	25%	0.248	0.195	0.191	0.211
	30%	0.289	0.237	0.234	0.253
	35%	0.338	0.285	0.281	0.301
ARTEX-MSA	25%	0.412	0.370	0.358	0.380
	30%	0.473	0.440	0.427	0.447
	35%	0.510	0.480	0.465	0.485

Table 4.4: *FRESA Scores of ARTEX-MSA over AMIS-Dataset30*

original document with FRESA. Results in Table 4.4 show that for all CRs, ARTEX-MSA performs better than the baseline system almost doubling its FRESA scores. These results are supported by the fact that the segments that ARTEX-MSA selects from a transcript to generate the corresponding summary are equally distributed over all the transcript (Table 4.3). Which impacts the performance of ARTEX-MSA, allowing it to cover more informative segments that may be in the middle or last part of the transcript.

4.2.3 Conclusion

In this section we conducted a study to analyze the performance of ARTEX-MSA over automatic transcripts. Selected transcripts were composed of a set of 30 samples from the *AMIS-Dataset* which were segmented with our SBD system presented in Section 3.1.1. Different from Section 4.1, where we summarized a set of documents from the Wikipedia and newspapers, transcripts from the *AMIS-Dataset* present the following extra complications:

1. Ideas are expressed different from well written text, causing longer and redundant SUs.
2. Domino effect errors produced by the concatenation of the ASR system and the SBD system (Figure 4.2).
3. Information distribution may not be the same over all the transcripts.

The lightweight approach that ARTEX-MSA follows to create extractive summaries is an advantage when facing with this complications. Given that ARTEX-MSA does not perform a detailed syntactic analysis, it is less susceptible to transcription and SBD errors. It also showed to select equally distributed segments along the transcript with a small preference for those segments in the middle section of the transcript. This way the resulting summary is able to cover all the transcript.

Chapter 5

Audio-based Multimedia Summarization

Contents

In a multimedia summarization context, documents do not limit to text uniquely. Depending of its source type, it is possible to take advantage of the multimedia nature of the document and drive different types of summaries (Figure 1.3). Each summarization approach focalizes on different aspects of the document, providing complementary summaries. In this chapter we explore multimedia summarization. Specifically, we analyze the summarization approaches around audio documents and how Information Retrieval (IR) techniques may help producing more informative audio summaries.

Automatic summarization of an audio document can be performed with the three following approaches: directing the summary using only audio features (Maskey and Hirschberg, 2005, 2006; Duxans et al., 2009; Zlatintsi et al., 2012), extracting the text inside the audio signal and directing the summarization process using textual methods (Taskiran et al., 2006; Christensen et al., 2008; Rott and Červa, 2016) and a hybrid approach which consists of a mixture of the first two (Zechner, 2003; Zlatintsi et al., 2015; Szaszák et al., 2016). Each approach has advantages and disadvantages with respect to the others.

Using only audio features for creating a summary has the advantage of being totally transcript independent, which is really useful when automatic or manual transcripts are not available. Nevertheless, this may also be a problem given that the summary is based only on how things are said and gives no importance of the informative content inside the document. By contrast, directing the summary with textual methods benefits from the information contained within the text, dealing to more informative summaries; however, transcripts may not be available in some cases. Finally, using both audio features and textual methods can boost the summary quality; still, disadvantages of both approaches are present.

The method we propose in this chapter consists of a hybrid approach during training phase while text independent during summary creation. It resides on using textual information to learn an informativeness representation based on probability distribution divergences that standard audio-based multimedia summarizers do not consider. During the summarization process this representation is used to obtain an informativeness score without a textual representation of the audio signal to summarize. To our knowledge, probability distribution divergences have not been used for audio summarization.

This chapter is organized as follows. In Section 5.1 we explain how the probability distribution divergence is used over an audio-based multimedia summarization framework and we describe in detail our summarization proposal. Next, in Section 5.2 we introduce a second audio summarizer based purely on audio features. Finally, in Section 5.3 we present and discuss the results obtained of both summarization strategies over a sample form the *AMIS-Dataset*.

5.1 Probability Distribution Divergence for Audio Summarization

Divergence is defined by Manning and Schütze (1999) as a function which estimates the difference between two probability distributions. In the framework of Automatic Text Summarization (ATS) evaluation, Louis and Nenkova (2008, 2009); Saggion et al. (2010); Torres-Moreno et al. (2010) have used divergence based measures such as Jensen-Shannon (JS) and Kullback-Leibler (KL) to compare the probability distribution of words between automatically produced summaries and their sources. Extractive summarization based on the divergence of probability distributions has been discussed in Louis and Nenkova (2008) and a summarization method (DIVTEX) has been proposed in Torres-Moreno (2014).

In this section we present our method based on probability distribution divergences to create audio summaries with an extractive summarization approach. It aims to select the most informative audio segments of the source signal until a time threshold is reached. A training phase is in charge of learning an informativeness model that maps a set of several audio features of an audio segment to an informativeness value. During this phase and after a preprocessing step, informativeness values are obtained by com-

puting the divergence between the documents of a big dataset and the corresponding documents segments. When a summary is to be created, the selected document is first preprocessed and segmented, then our method uses the trained informativeness model to map the audio features of the segments to their corresponding informativeness values and then rank their pertinence.

5.1.1 Audio Signal Reprocessing

During the preprocessing step, the input audio signal is split into background and foreground channels. This process is normally used on music records for separating vocals and other sporadic signals from accompanying instrumentation. Rafii and Pardo (2012) performed this separation for identifying recurrent elements by looking for similarities instead of periodicities. Their approach is useful for song records where repetitions happen intermittently or without a fixed period. However, we found that applying the same method to newscasts and reports audio files made much easier to segment them using only the background signal. We assume this phenomenon is due to the fact that newscasts and reports are heavily edited with a low volume of background music playing while the journalist speaks (background) and louder music/noises for transitions (foreground).

The input audio signal is split into background and foreground channels based on Rafii and Pardo (2012). First, audio frames are compared using the cosine similarity; similar frames separated by at least two seconds are aggregated by taking their per-frequency median value to avoid being biased by local continuity. This is done to suppress non-repetitive deviations from the average spectrum and discard vocal elements. Next, assuming that both signals are additive, a pointwise minimum between the obtained frames and the original signal is applied to obtain a raw background filter. Then, a foreground and background time-frequency mask is derived from the raw background filter and the input signal with a soft mask operation. Finally, the foreground and background components are obtained by multiplying the time-frequency masks with the input signal.

5.1.2 Training Phase (Informativeness Model)

During this phase, an informativeness model which maps a set of 277 audio features with informativeness values from a big audio dataset and their transcripts is learned. Informativeness values correspond to the JS divergences between the segmented transcripts and their source. The JS divergence $D_{JS}(P||Q)$ between a segment Q and its source P is implemented as defined by Louis and Nenkova (2009) and Torres-Moreno et al. (2010):

$$D_{JS}(P||Q) = \frac{1}{2} \sum_{w \in P} \left[P_w \cdot \log_2 \left(\frac{2P_w}{P_w + Q_w} \right) + Q_w \cdot \log_2 \left(\frac{2Q_w}{P_w + Q_w} \right) \right] \tag{5.1}$$

where

$$P_w = \frac{C_w^P + \delta}{|P| + \delta \cdot \beta} \qquad \text{and} \qquad Q_w = \frac{C_w^Q + \delta}{|Q| + \delta \cdot \beta};$$

$C_w^{(P|Q)}$ is the frequency of word w over P or Q. To avoid shifting the probability mass to unseen events, the scaling parameter δ is set to 0.0005. $|P|$ and $|Q|$ correspond to the number of tokens on P and Q. Finally $\beta = 1.5|V|$, where $|V|$ is the vocabulary size on P.

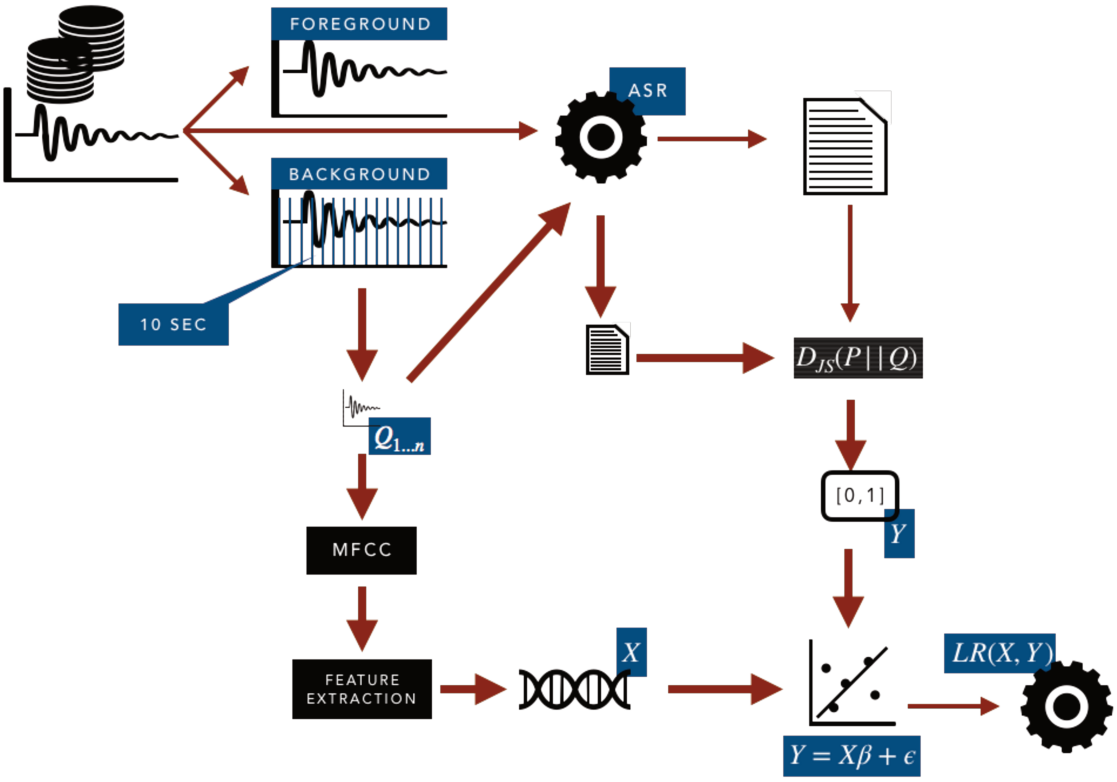

Figure 5.1: *Informativeness Model Training Scheme*

Figure 5.1 illustrates the whole informativeness model training phase. First, the preprocessing step described in Section 5.1.1 is applied for each sample P within the training dataset. Next, the background channel of the sample is split into 10 seconds length segments $Q_{1...n}$. Then, each segment Q is represented by 277 audio features X, where 275 correspond to 11 statistical values of 25 Mel-frequency Cepstral Coefficients (MFCCs) and the other two correspond to the number of frames in the segment and its starting time. The 11 statistical values can be seen in Table 5.1, where ϕ' and ϕ'' corresponds to the first and second MFCCs derivatives.

Feature	MFCC		
	ϕ	ϕ'	ϕ''
Min	•		
Max	•		
Median	•		
Mean	•	•	•
variance	•	•	•
Skewness	•		
Kurtosis	•		

Table 5.1: *MFCC-based Statistical Values*

- **Min**: The smallest value.
- **Max**: The biggest value.
- **Median**: The median from all values.
- **Mean**: The mean from all values.
- **Variance**: The variance of all values.
- **Skewness**: The symmetry within the distribution of the values.
- **Kurtosis**: The tail-heaviness of the distribution of values.

To obtain the informativeness score Y of each segment Q, the JS divergence D_{JS} is computed between the segment Q_i and its source P as defined in Equation 5.1. Finally a linear least squares regression model $LR(X, Y)$ is trained with all segments audio features X and informativeness scores Y. All audio processing and feature extraction is performed with the Librosa library [1] (McFee et al., 2015).

Mel-frequency Cepstral Coefficients

The MFCCs (Bridle and Brown, 1974; Mermelstein, 1976) are short-term spectral-based feature used in automatic speech and speaker recognition.

The process of creating MFCCs features as described by Logan et al. (2000) is the following:

1. Divide signal into short frames.
2. Compute the Discrete Fourier Transform of each frame
3. Retain only the logarithm of the amplitude spectrum.

[1] https://librosa.github.io/librosa/index.html

4. Convert to Mel-spectrum
5. Take the Discrete Cosine Transformation

5.1.3 Audio Summary Creation

When a summary is to be created, only the audio signal and the trained informativeness model described in Section 5.1.2 are needed. Figure 5.2 illustrates the full process to obtain an extractive summary from an audio document P. First, the preprocessing step described in Section 5.1.1 is applied over P. After the background signal is isolated from the main signal, 25 MFCCs are computed to characterize the audio stream and a temporally-constrained agglomerative clustering routine is used to partition it into k contiguous segments $Q_{1...k}$, defined as:

$$k = \frac{P_{length}}{60} \times 20 \tag{5.2}$$

where P_{length} corresponds to the length in seconds of P. Then, each segment Q_i is represented by the same 277 audio features described in Section 5.1.2 to obtain X_{Q_i}. Next, the informativeness LR_{Q_i} of each segment $Q_i \in P$ is predicted with the linear model $LR(X_{Q_i}, Y_{Q_i})$. Finally, a score S_{Q_i} is computed for each segment $Q_1...Q_k$ as follows:

$$S_{Q_i} = e^{1-LR_{Q_i}} \tag{5.3}$$

The audio-based summary is created by choosing those segments that contain the highest S_{Q_i} scores until the desired Compression Ratio (CR) is reached; then they are placed in order of appearance.

5.2 Audio Features for Audio Summarization

We propose a second audio-based summarizer to contrast performance and functionality of our system based on informativeness. In this second approach no training phase is needed and no big audio dataset is required. Figure 5.3 illustrates the full summarization process for this method.

Similar to our audio-based summarizer based on informativeness, the preprocessing step described in Section 5.1.1 is first applied over the document P to be summarized. Next, 25 MFCCs are computed over the background signal and following Equation 5.2, it is divided into k contiguous segments $Q_{1...k}$ with a temporally-constrained agglomerative clustering routine. Each segment Q_i is then represented by the same 277 audio features described in Section 5.1.2 to obtain X_{Q_i}. Finally, for each segment $Q_1 \dots Q_k$, a score S_{Q_i} is computed as follows:

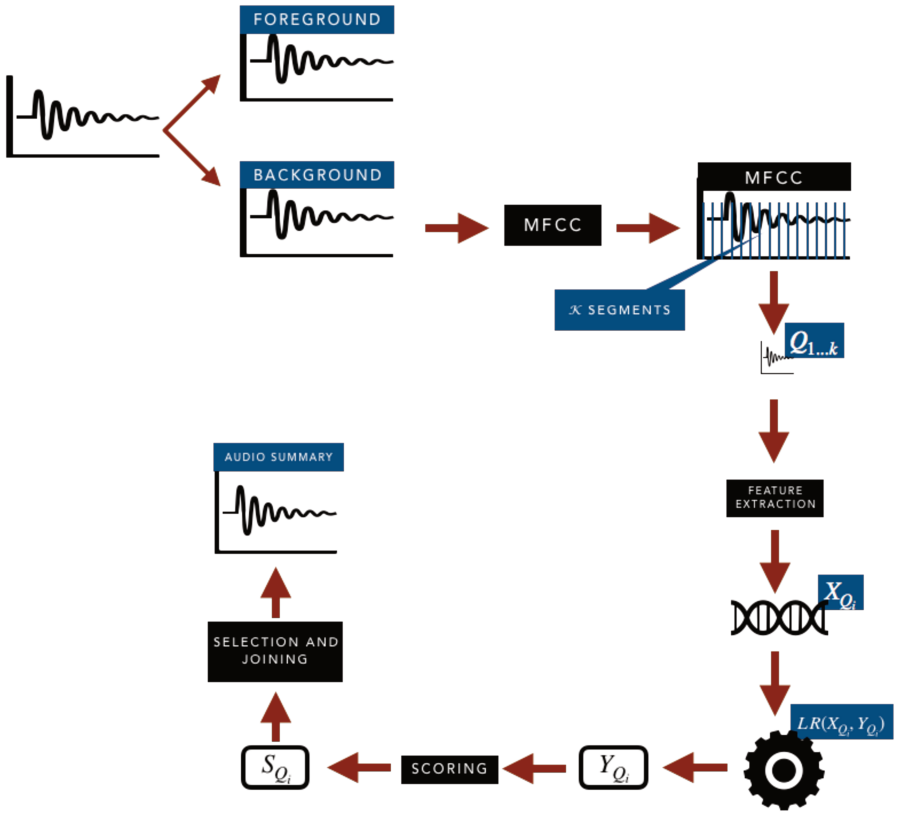

Figure 5.2: *Audio Summary Creation Scheme Based on Informativeness Model*

$$S_{Q_i} = \frac{1}{1+e^{-(\Delta_{t_i}-5)}} \times \frac{|Q_i|}{|P|} \times e^{-\frac{t_{Q_i}}{\Delta_{t_i}}} \tag{5.4}$$

where $\Delta_{t_i} = t_{Q_i+1} - t_{Q_i}$, being t_{Q_i} the starting time of the segment Q_i and t_{Q_i+1} the starting time of Q_{i+1}. $|Q_i|$ and $|P|$ correspond to the length in seconds of the segment Q_i and the source P respectively.

5.3 Experimental Evaluation

We were interested in seeing how segments that compose audio-based summaries were distributed over the original documents and in knowing which one of the presented method provides the most informative summaries. For this, we proposed a comparative experimental evaluation between the summarization methods described in sections 5.1 and 5.2.

To train our audio summarizer based on informativeness we employed all the samples from the *AMIS-Dataset*. Transcripts needed to compute JS divergences were produced with the Automatic Speech Recognition (ASR) system described on Jouvet et al.

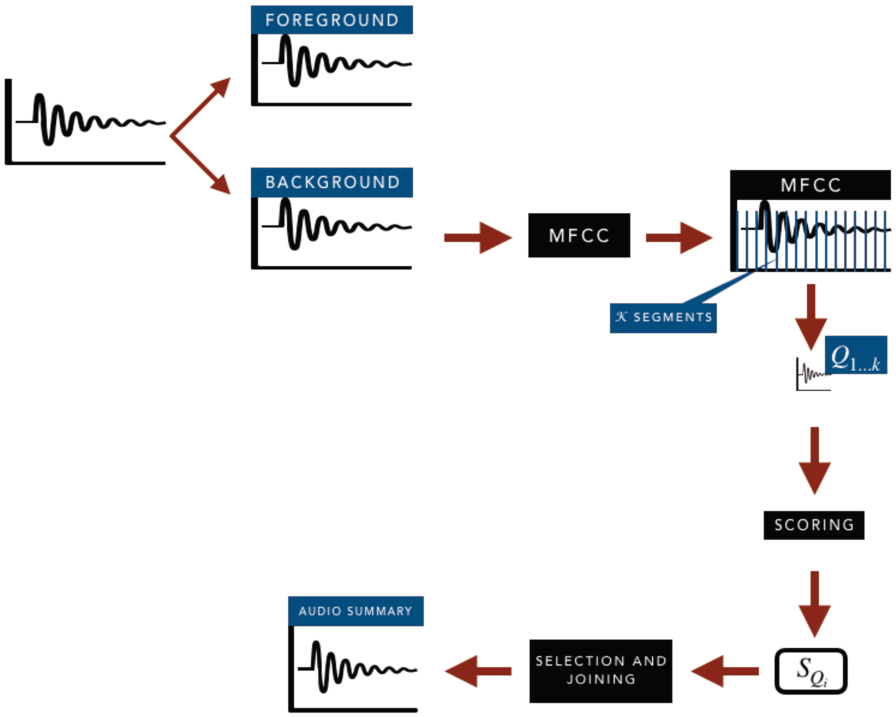

Figure 5.3: *Audio Summary Creation Scheme Based on Audio Features*

(2018). After transcription process we applied a stemming phase to reduce inflected and derived words to their root. During audio summary creation we focused on a set of 10 English YouTube videos with similar characteristics than those in the *AMIS-Dataset*. The length of this 10 samples as well as their number of segments are shown in the left section from Table 5.3. As it can be seen, samples' length varies between 1m42s and 9m45s with an average length equal to 5m18s.

5.3.1 Evaluation Metric

Based on the evaluation scale proposed by Rott and Červa (2016) to evaluate the readability and relevance of a text summary, we used the subjective scale shown in Table 5.2 to measure the informativeness of audio summaries. The same scale was used to measure the informativeness of a generated audio summary as a whole and to measure the informativeness of each one of the segments that compose the summary.

To perform evaluation we gave a group of five evaluators with the web interface shown in Figure 5.4 and asked them to evaluate the set of automatically created summaries. The web interface provided evaluators with the freedom of choosing which summary to evaluate with no time restriction; the only constraint we gave to them was to use always the same evaluation label. The interface is divided in the following three main sections:

Score	Explanation
5	Full informative
4	Mostly informative
3	Half informative
2	Quite informative
1	Not informative

Table 5.2: *Audio Summarization Evaluation Scale*

			Audio Features for Audio Summarization				Probability Distribution Divergence for Audio Summarization				
Sample	**Length**	**Segments**	**Selected Segments**	**Segments Length**	**Full Score**	**Segments Score**	**Selected Segments**	**Segments Length**	**Clustered Segments**	**Full Score**	**Segments Score**
1	3m19s	65	8	9.81s	4.20	2.90	25	2.88s	5	3.00	2.12
2	5m21s	106	13	9.44s	3.50	2.78	63	1.79s	11	3.00	1.62
3	2m47s	55	5	12.21s	3.80	3.76	9	7.23s	4	2.00	2.22
4	1m42s	33	5	9.22s	3.60	2.95	10	3.59s	4	3.00	2.30
5	8m47s	175	22	8,82s	4.67	3.68	50	4.34s	10	3.00	2.46
6	9m45s	184	30	6.97s	4.00	2.49	52	3.96s	17	4.00	1.87
7	5m23s	107	8	15.02s	3.20	3.75	43	2.66s	7	2.00	1.35
8	6m23s	127	20	6.97s	3.75	2.84	37	3.65s	12	3.00	2.05
9	7m35s	151	18	9.58s	3.75	3.19	78	2.09s	13	3.00	1.71
10	2m01s	39	4	10.77s	2.75	2.63	9	4.97s	5	3.00	2.33
Mean	5m18s	104.20	13.30	9.89s	3.72	3.12	37.60	3.72s	8.80	2.90	2.00

Table 5.3: *Audio Summarization Performance over Complete Summaries and Summary Segments*

- **ORIGINAL AUDIO DOCUMENT**: It allows to select the desired audio file and to listen to it.
- **COMPLETE SUMMARY EVALUATION**: It presents the audio summary and allows to rate it following the scale from Table 5.2.
- **SUMMARY SEGMENTS EVALUATION**: It lists all the segments that compose the audio summary, allowing the evaluator to rate them independently following the scale from Table 5.2.

5.3.2 Results

We ran both summarization methods over the 10 English videos to perform audio summaries with a CR of 35%. Evaluation was performed over the complete audio summaries as well as over each summary segment given our interest in measuring the informativeness of the summaries but also the informativeness of each one of the segments that compose the summaries.

Central region of Table 5.3 presents the results of the audio summarization method based on audio features (Section 5.2). "Selected Segments" corresponds to the number of segments that were used by the method to create the summary. "Full Score" refers to the average score of complete audio summaries for each sample, whereas "Seg-

Figure 5.4: *Summary Evaluation Web Interface*

ments Score" to the average score of the segments that belong to each summary. "Full Score" and "Segments Score" represent different things and seem to be correlated. "Full Score" quantifies the informativeness of all the summary as a whole while "Segments Score" represents the summary quality in terms of the informativeness of each one of its segments. Right region of Table 5.3 presents the results of the probability distribution divergence audio summarization method. "Selected Segments", "Full Score" and "Segments Score" correspond to the same elements than the central region, yet an extra column is present. "Clustered Segments" refers to number of segments groups after adjacent segments had been joined.

Summaries produced by the audio features method have in average 13.30 segments; a much smaller value compared to the 37.60 segments from the probability distribution divergence model method. This means that the scoring function of the audio features method favors longer segments, which may be or not an advantage depending

of the content of the segment. A long informative segment is beneficial to a summary; notwithstanding, a long segment that lacks from information contributes nothing to the summary.

Segment distribution over summaries can be visualized in figures 5.5 and 5.6. The long blue rectangles correspond to the complete audio samples while the lighter rectangles correspond to each one of the segments that compose the summary. Black vertical lines divide the audio samples in half. Length of the complete audio samples and summary segments is represented by the rectangles' width. Score of each segment is represented by its height. Summary segments from the audio features method are visually longer than those from the probability distribution divergence model summarizer. This observation is supported by the "Segments Length" columns of Table 5.3 where the average segments length is 9.89s for the audio features method and 3.72s for the probability distribution divergence model summarizer. This second method retrieves a lot of short segments, but when adjacent segments are grouped, the average number of segments within the summaries is reduced to 8.80.

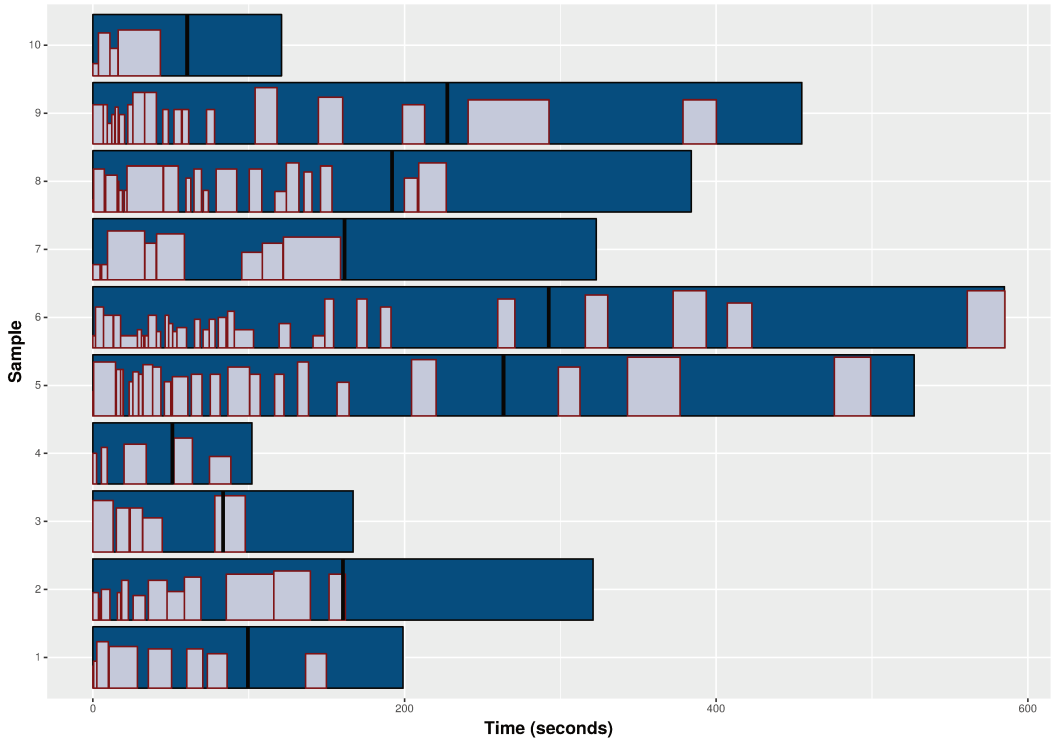

Figure 5.5: *Summary Segments Distribution for the Audio Features Summarizer*

With regard to the audio features method, average scores of complete summaries oscillate between 2.75 and 4.67; except from Sample#10, all produced summaries are at least "Half informative". On the other hand, values corresponding to the average segments scores oscillate between 2.49 and 3.76; in this case, only the segments of 40% of the summaries are considered to be at least "Half informative". An interesting case is Sample#6, which according to its "Full Score" is "Mostly informative" but has the low-

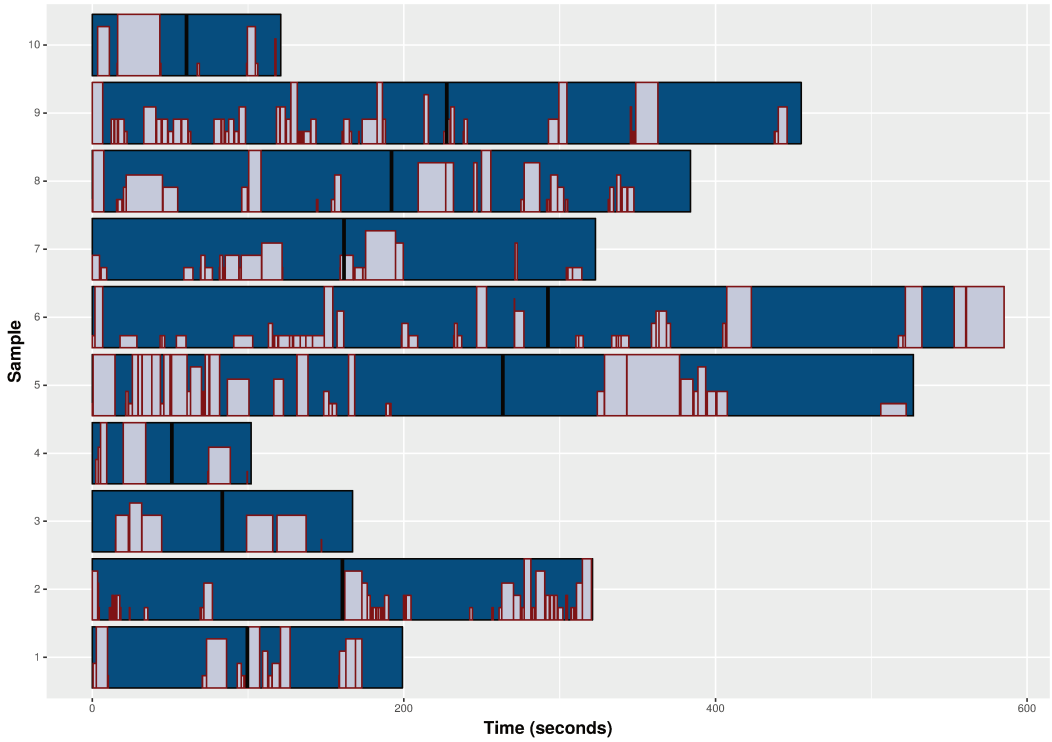

Figure 5.6: *Summary Segments Distribution for the Informativeness Model Summarizer*

est "Average Score" of all samples. This difference is given because 67% of its segments has an informativeness score smaller than 3, but it achieves to communicate almost all relevant information. It can be seen that the summaries produced by the audio features method have the majority of their segments clustered to the left. This behaviour is the result of the preference that the summarization technique gives to the first part of the audio stream, where in newscast is gathered the major part of the information. This may be a problem in cases where different topics are covered over the newscast (multi-topic newscast, interviews, round tables, reports, etc.) where relevant information is distributed all over the video. If a big amount of relevant segments are present in the first minutes of the newscast, the summarization algorithm will use all the space available for the summary very fast, discarding a large region of the audio stream. An example is the case of Sample#7 and Sample#10 which "Full Scores" are less than 3.50. Concerning Sample#5, a well distribution of its summary segments is observed. From its 22 segments, only 4 of them have an informativeness score less than 3, achieving the highest "Full Score" of all samples and a good "Average Score". Figures A.3 to A.12 show the average segments score of each sample.

Concerning the probability distribution divergence model summarizer, the 80% of "Full Scores" is at least "Half informative"; while average segments scores are considered only to be "Quite informative". This is expected given the short duration of the summary segments, which independently does not supply much information to the summary. Nevertheless, given that a lot of these short segments are adjacent, they

manage to create "Half informative" summaries. Sample#6 presents a similar behavior to the audio features summarizer. In this case, 77% of its segments have an informativeness score smaller than 3, but it achieves to be "Mostly informative". Compared to the audio features summarizer, in this case the summary segments are more equally distributed over the audio samples; however, the average segment length is 2.65 times shorter. It is interesting to observe the agglomerative behaviour the summary segments present. In almost all samples it is possible to observe regions of at least four adjacent segments, thus producing longer and more informative regions. A good example is Sample#6, where 13 of its 52 segments are clustered into a single segment. Detailed average segments scores for each sample can be seen from Figure A.13 to A.22.

5.4 Conclusion

In this chapter we presented an extractive audio-based multimedia summarizer which uses probability distribution divergences of transcripts and audio features to learn an informativeness model during training phase. During summary creation, only audio features are needed and the informativeness model is used to decide which segments should be included in the summary. We also proposed summarizer based only on audio features and exempt of any informativeness model.

Qualitative results of both methods over a small English dataset showed that both systems behave different. Segments from the audio features summarizer resulted to be long but most of them agglutinated in the first half of the audio stream. By contrast, summary segments from the probability distribution divergence model summarizer were short but well distributed.

The audio-based summarizer based on probability distribution divergences has been integrated into the Access Multilingual Information opinionS (AMIS) project as part of architecture SC2 (Smaïli et al., 2019).

Chapter 6

WiSeBE: Window-based Sentence Boundary Evaluation

Contents

The goal of Automatic Speech Recognition (ASR) is to transform spoken data into a written representation, thus enabling natural human-machine interaction (Yu and Deng, 2016) with further Natural Language Processing (NLP) tasks. Machine Translation (MT), Question Answering (QA), semantic parsing, Part-of-Speech (POS) tagging, sentiment analysis and Automatic Text Summarization (ATS); originally developed to work with formal written texts, can be applied over the transcripts made by ASR systems (Stevenson and Gaizauskas, 2000; Wang et al., 2010; Brum et al., 2016). However, before applying any of these NLP tasks a segmentation process called Sentence Boundary Detection (SBD) should be performed over ASR transcripts to reach a minimal syntactic information in the text.

Performance of a SBD system is normally measured by comparing the automatically segmented transcript against a single reference normally done by a human. But given a transcript, does it exist a unique reference? Or, is it possible that the same transcript could be segmented in five different ways by five different people in the same conditions? If so, which one is correct? And more important, how to fairly evaluate the automatically segmented transcript? These questions are the foundations of Window-based

Sentence Boundary Evaluation (WiSeBE), a new semi-supervised metric for evaluating SBD systems based on multi-reference (dis)agreement.

This chapter is organized as follows: in Section 6.1 we formally describe WiSeBE and the *WiSeBE-score*. Then, in Section 6.2 we evaluate two SBD systems following a multi-reference strategy, where we compare a standard SBD evaluation with WiSeBE. Finally, in this same section we discuss and provide further analysis with respect to WiSeBE.

6.1 Window-based Sentence Boundary Evaluation

Window-based Sentence Boundary Evaluation (WiSeBE) is a semi-automatic multi-reference sentence boundary evaluation protocol inspired on the works of Lin (2004) and Nenkova and Passonneau (2004) for text summaries evaluation. It considers the performance of a candidate segmentation over a set of segmentation references and the agreement between those references.

Let $\mathbf{R} = \{R_1, R_2, ..., R_m\}$ be the set of all available references given a transcript $T = \{t_1, t_2, ..., t_n\}$, where t_j is the j^{th} word in the transcript; a reference R_i is defined as a binary vector in terms of the Semantic Unit (SU) boundaries in T.

$$R_i = \{b_1, b_2, ..., b_n\} \tag{6.1}$$

where

$$b_j = \begin{cases} 1 & \text{if } t_j \text{ is a boundary} \\ 0 & \text{otherwise} \end{cases}$$

Given a transcript T, the candidate segmentation C_T is defined similar to R_i.

$$C_T = \{b_1, b_2, ..., b_n\} \tag{6.2}$$

where

$$b_j = \begin{cases} 1 & \text{if } t_j \text{ is a boundary} \\ 0 & \text{otherwise} \end{cases}$$

6.1.1 General Reference and Agreement Ratio

A General Reference (R_G) is then constructed to calculate the agreement ratio between all references. It is defined by the boundary frequencies of each reference $R_i \in \mathbf{R}$.

$$R_G = \{d_1, d_2, ..., d_n\} \tag{6.3}$$

where

$$d_j = \sum_{i=1}^{m} t_{ij} \quad \forall t_j \in T, \quad d_j = [0, m] \tag{6.4}$$

An Agreement Ratio ($R_{G_{AR}}$) is needed to get a numerical value of the distribution of SU boundaries over **R**.

$$R_{G_{AR}} = \frac{R_{G_{PB}}}{R_{G_{HA}}} \tag{6.5}$$

where $R_{G_{PB}}$ corresponds to the ponderated common boundaries of R_G while $R_{G_{HA}}$ to its hypothetical maximum agreement.

$$R_{G_{PB}} = \sum_{j=1}^{n} d_j \left[d_j \geq 2 \right] \tag{6.6}$$

$$R_{G_{HA}} = m \times \sum_{d_j \in R_G} 1 \left[d_j \neq 0 \right] \tag{6.7}$$

A value of $R_{G_{AR}}$ close to 0 means a low agreement between references in **R**, while $R_{G_{AR}} = 1$ means a perfect agreement ($\forall R_i \in \mathbf{R}, R_i = R_{i+1} | i = 1, ..., m-1$) in **R**.

6.1.2 Window-boundaries Reference

In Section 2.3 we discussed how disfluencies complicate SU segmentation. In a multi-reference environment this causes disagreement between references around a same SU boundary. The way WiSeBE handle disagreements produced by disfluencies is with a Window-boundaries Reference (R_W) defined as:

$$R_W = \{w_1, w_2, ..., w_p\} \tag{6.8}$$

where each window w_k considers one or more boundaries d_j from R_G with a window separation limit equal to R_{W_l}.

$$w_k = \{d_j, d_{j+1}, d_{j+2}, ...\} \tag{6.9}$$

The algorithm used to create R_W is described in Algorithm 1.

Algorithm 1 Window-boundaries Reference Creation

Input: R_G (general reference), R_{W_l} (window separation limit)
$w \leftarrow \emptyset$ (temporal window)
$R_W \leftarrow \emptyset$ (window boundaries)
$last_b \leftarrow 0$ (last boundary intex)
for each boundary b_i in R_G **do**
  if i - $last_b > R_{W_l}$ and $W > 0$ **then**
    $R_W.add(w)$
    $w \leftarrow \emptyset$
  end if
  if b_i is different from 0 **then**
    $w.add(i)$
    $last_b \leftarrow i$
  end if
  if $w > 0$ **then**
    $R_W.add(w)$
  end if
end for
Return: R_W

6.1.3 *WiSeBE-score*

WiSeBE-score is a normalized score dependent of 1) the performance of C_T over R_W and 2) the agreement between all references in **R**. It is defined as:

$$\textit{WiSeBE-score} = \textit{F1-score}_{R_W} \times R_{G_{AR}} \quad \textit{WiSeBE-score} = [0, 1] \tag{6.10}$$

where $\textit{F1-score}_{R_W}$ corresponds to the harmonic mean of *Precision* and *Recall* of C_T over R_W, while $R_{G_{AR}}$ is the agreement ratio defined in Equation 6.5. $R_{G_{AR}}$ can be interpreted as a scaling factor; a low value will penalize the overall *WiSeBE-score* given the low agreement between references. By contrast, for a high agreement in **R** ($R_{G_{AR}} \approx 1$), $\textit{WiSeBE-score} \approx \textit{F1-score}_{R_W}$.

$$\textit{F1-score}_{R_W} = 2 \times \frac{\textit{Precision}_{R_W} \times \textit{Recall}_{R_W}}{\textit{Precision}_{R_W} + \textit{Recall}_{R_W}} \tag{6.11}$$

$$\textit{Precision}_{R_W} = \frac{\sum_{b_j \in C_T} 1 \quad [b_j = 1, b_j \in w \quad \forall w \in R_W]}{\sum_{b_j \in C_T} 1 \quad [b_j = 1]} \tag{6.12}$$

$$\textit{Recall}_{R_W} = \frac{\sum_{w_k \in R_W} 1 \quad [w_k \ni b \quad \forall b \in C_T]}{p} \tag{6.13}$$

Equations 6.12 and 6.13 describe $\textit{Precision}_{R_W}$ and $\textit{Recall}_{R_W}$ of C_T over R_W. $\textit{Precision}_{R_W}$ is the number of boundaries b_j inside any window w_k from R_W divided by the total

number of boundaries b_j in C_T. $Recall_{R_W}$ corresponds to the number of windows w with at least one boundary b divided by the number of windows w in R_W.

6.2 Evaluating with WiSeBE

To exemplify the Window-based Sentence Boundary Evaluation (WiSeBE) evaluation protocol we evaluated and compared the performance of two different Sentence Boundary Detection (SBD) systems over a set of YouTube video transcripts in a multi-reference environment. The first system (S_1) corresponds to CNN-B, described in Section 3.1.1 with the English model trained in Section 3.1.3. The second system (S_2) corresponds to a bidirectional Recurrent Neural Network (RNN) model with attention mechanism for boundary detection proposed by Tilk and Alumäe (2016).

6.2.1 Dataset

We performed evaluation with the same video samples from the *AMIS-Dataset* that were used to perform audio-based multimedia summarization in Section 5.3. This videos cover different topics like technology, human rights, terrorism and politics with a length variation between 2 and 10 minutes in different formats like newscasts, interviews, reports and round tables. During the transcription phase we opted for a manual transcription process because we observed that using transcripts from an Automatic Speech Recognition (ASR) system will difficult in a large degree the manual segmentation process.

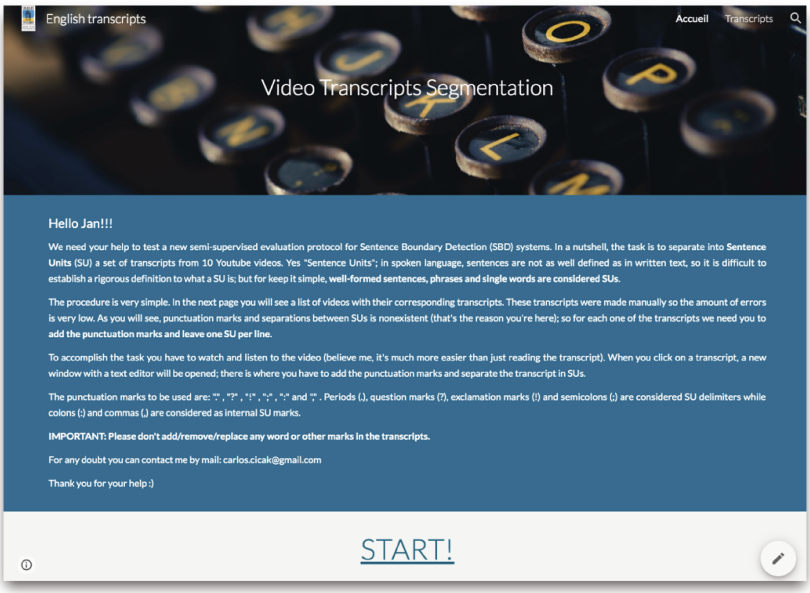

Figure 6.1: *Sentence Boundary Detection Web Interface (Instructions)*

We provided a group of three annotators (ref_1, ref_2, ref_3) with a web interface to perform the manual SBD annotation work. All annotators were advanced or native English speakers and had experience in linguistic and annotation tasks. Clear instructions of how segmentation was needed to be performed, including the Semantic Unit (SU) concept and how punctuation marks were going to be taken into account are shown in Figure 6.1. Periods (.), question marks (?), exclamation marks (!) and semicolons (;) were considered SU delimiters (boundaries) while colons (:) and commas (,) were considered as internal SU marks.

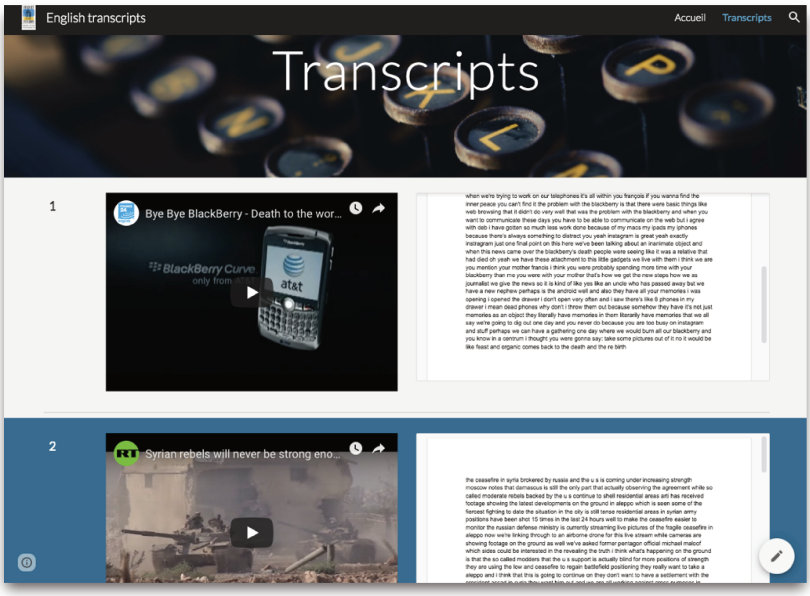

Figure 6.2: *Sentence Boundary Detection Web Interface (Overview)*

Figure 6.2 displays each one of the 10 samples, providing a link to the video and the manual transcript. The text editor where annotators were asked to segment the transcript is shown in Figure 6.3.

Left section of Table 6.1 presents the number of words per transcript within the dataset, while middle section the number of segments per transcript and reference. An interesting remark is that ref_3 assigns about 43% less boundaries than the mean of the other two references.

6.2.2 Results

We ran both systems (S_1 & S_2) over the manually transcribed videos obtaining the number of boundaries shown in the right section of Table 6.1. In general, it can be seen that S_1 predicts 27% more segments than S_2. This difference can affect the performance of S_1, increasing its probabilities of false positives.

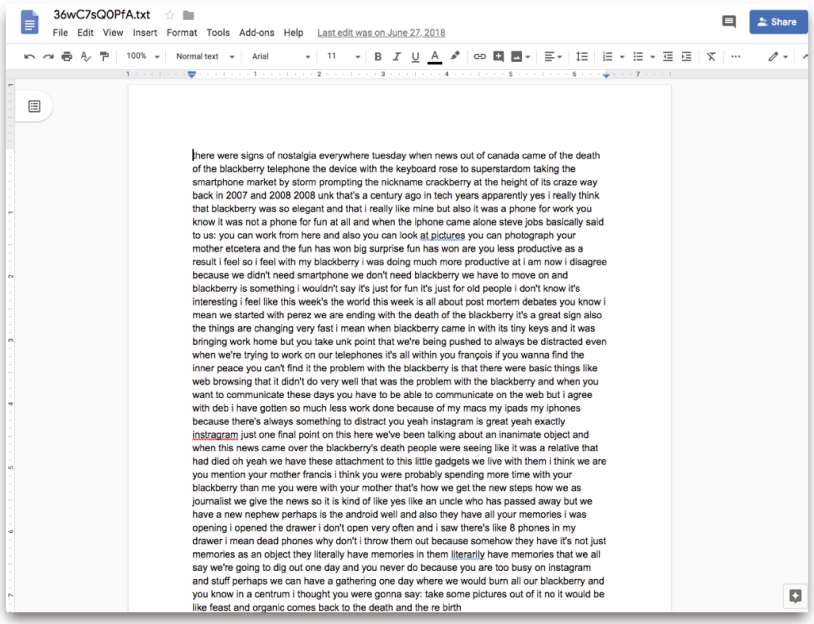

Figure 6.3: *Sentence Boundary Detection Web Interface (Segmentation)*

Independent Multi-reference Evaluation

Table 6.2 condenses the performance of both systems evaluated against each one of the references independently. If we focus on *F1-score*, performance of both systems varies depending of the reference. For ref_1, S_1 was better in 5 occasions compared to S_2; S_1 was better in only 2 occasions for ref_2; S_1 overperformed S_2 in 3 occasions concerning ref_3 and in 4 occasions for the mean values (**bold scores**). Also from Table 6.2 we can observe that ref_1 has a bigger similarity to S_1 in 6 occasions compared to other two references, while ref_2 is more similar to S_2 in 7 transcripts (underlined scores). From the mean *F1-scores* it can be concluded that in average S_2 had a better performance segmenting the dataset compared to S_1, obtaining a *F1-score* equal to 0.510. But, what about the complexity of the dataset? Regardless all references have been considered, nor agreement or disagreement between them has been taken into account.

WiSeBE Evaluation

All values related to the *WiSeBE-score* are displayed in Table 6.3. The smallest Agreement Ratio ($R_{G_{AR}}$) between references corresponds to Sample#8 while the biggest to Sample#5. The lower the $R_{G_{AR}}$, the bigger the penalization *WiSeBE-score* will present. A good example is S_2 for Sample#4 where $\textit{F1-score}_{R_W}$ reaches a value of 0.800, but after considering $R_{G_{AR}}$ the *WiSeBE-score* drops to 0.462.

It is feasible to think that if all references are taken into account at the same time during evaluation ($\textit{F1-score}_{R_W}$), the score will be bigger compared to an average of in-

Sample	Words	Manual Segmentation				Automatic Segmentation	
		ref_1	ref_2	ref_3	**Mean**	S_1	S_2
1	621	38	33	23	31	53	38
2	731	42	42	20	35	38	37
3	338	17	16	10	14	15	12
4	236	14	11	6	10	13	11
5	644	55	54	39	49	54	36
6	1 602	87	98	39	75	108	92
7	1 540	109	92	76	92	106	86
8	1 194	65	72	30	56	70	46
9	903	55	51	29	45	71	53
10	271	20	16	9	15	11	13
Total	8 080	502	485	281	422	539	424

Table 6.1: *Manual and Automatic Segmentation*

Sample	System	ref_1			ref_2			ref_3			**Mean**		
		P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
1	S_1	0.396	0.553	0.462	0.377	0.606	0.465	0.264	0.609	0.368	0.346	0.589	0.432
	S_2	0.474	0.474	**0.474**	0.474	0.545	**0.507**	0.368	0.609	**0.459**	0.439	0.543	**0.480**
2	S_1	0.605	0.548	**0.575**	0.711	0.643	**0.675**	0.368	0.700	**0.483**	0.561	0.630	**0.578**
	S_2	0.595	0.524	0.557	0.676	0.595	0.633	0.351	0.650	0.456	0.541	0.590	0.549
3	S_1	0.333	0.294	0.313	0.267	0.250	0.258	0.200	0.300	0.240	0.267	0.281	0.270
	S_2	0.417	0.294	**0.345**	0.417	0.313	**0.357**	0.250	0.300	**0.273**	0.361	0.302	**0.325**
4	S_1	0.615	0.571	0.593	0.462	0.545	0.500	0.308	0.667	0.421	0.462	0.595	0.505
	S_2	0.909	0.714	**0.800**	0.818	0.818	**0.818**	0.455	0.833	**0.588**	0.727	0.789	**0.735**
5	S_1	0.630	0.618	**0.624**	0.593	0.593	**0.593**	0.481	0.667	**0.560**	0.568	0.626	**0.592**
	S_2	0.667	0.436	0.527	0.611	0.407	0.489	0.500	0.462	0.480	0.593	0.435	0.499
6	S_1	0.491	0.541	**0.515**	0.454	0.563	0.503	0.213	0.590	0.313	0.386	0.565	0.443
	S_2	0.500	0.469	0.484	0.522	0.552	**0.536**	0.250	0.590	**0.351**	0.423	0.537	**0.457**
7	S_1	0.594	0.578	**0.586**	0.462	0.533	0.495	0.406	0.566	0.473	0.487	0.559	0.518
	S_2	0.663	0.523	0.585	0.558	0.522	**0.539**	0.465	0.526	**0.494**	0.562	0.524	**0.539**
8	S_1	0.443	0.477	0.459	0.514	0.500	0.507	0.229	0.533	0.320	0.395	0.503	0.429
	S_2	0.609	0.431	**0.505**	0.652	0.417	**0.508**	0.370	0.567	**0.447**	0.543	0.471	**0.487**
9	S_1	0.437	0.564	0.492	0.451	0.627	0.525	0.254	0.621	0.360	0.380	0.603	**0.459**
	S_2	0.623	0.600	**0.611**	0.585	0.608	**0.596**	0.321	0.586	**0.414**	0.509	0.598	0.541
10	S_1	0.818	0.450	**0.581**	0.818	0.450	**0.581**	0.455	0.556	**0.500**	0.697	0.523	**0.582**
	S_2	0.692	0.450	0.545	0.615	0.500	0.552	0.308	0.444	0.364	0.538	0.465	0.487
Mean	S_1	—		0.520	—		0.510	—		0.404	—		0.481
	S_2	—		**0.543**	—		**0.554**	—		**0.433**	—		**0.510**

Table 6.2: *Independent Multi-reference Evaluation Results. P: Precision; R: Recall; F1: F1-score*

Sample	System	$F1\text{-}score_{mean}$	$F1\text{-}score_{R_W}$	$R_{G_{AR}}$	$WiSeBE\text{-}score$
1	S_1	0.432	0.495	0.691	0.342
	S_2	**0.480**	0.513		**0.354**
2	S_1	**0.578**	0.659	0.688	**0.453**
	S_2	0.549	0.595		0.409
3	S_1	0.270	0.303	0.684	0.207
	S_2	**0.325**	0.400		**0.274**
4	S_1	0.505	0.593	0.578	0.342
	S_2	**0.735**	0.800		**0.462**
5	S_1	**0.592**	0.614	0.767	**0.471**
	S_2	0.499	0.500		0.383
6	S_1	0.443	0.550	0.541	**0.298**
	S_2	**0.457**	0.535		0.289
7	S_1	0.518	0.592	0.617	0.366
	S_2	**0.539**	0.606		**0.374**
8	S_1	0.429	0.494	0.525	0.259
	S_2	**0.487**	0.508		**0.267**
9	S_1	0.459	0.569	0.604	0.344
	S_2	**0.541**	0.667		**0.403**
10	S_1	**0.582**	0.581	0.619	**0.359**
	S_2	0.487	0.545		0.338
Mean	S_1	0.481	0.545	0.631	0.344
	S_2	**0.510**	0.567		**0.355**

Table 6.3: *WiSeBE Evaluation Results*

dependent evaluations ($F1\text{-}score_{mean}$); however this is not always true. This is the case of S_1 for Sample#10, which present a slight decrease for $F1\text{-}score_{R_W}$ compared to $F1\text{-}score_{mean}$. An important remark is the behavior of S_1 and S_2 concerning Sample#6. If evaluated without considering any (dis)agreement between references ($F1\text{-}score_{mean}$), S_2 overperforms S_1; nevertheless, this is inverted once the systems are evaluated with WiSeBE.

6.2.3 Discussion

$R_{G_{AR}}$ and Fleiss' kappa Correlation

In Section 6.1.3 we described the *WiSeBE-score* and how it relies on the $R_{G_{AR}}$ value to scale the performance of C_T over R_W. $R_{G_{AR}}$ can intuitively be considered an agreement value over all elements of **R**. To test this hypothesis, we computed the Pearson Correlation Coefficient (PCC) (Pearson, 1895) between $R_{G_{AR}}$ and the Fleiss' kappa (Fleiss, 1971) (κ_R) of each sample within the dataset.

A linear correlation between $R_{G_{AR}}$ and κ_R can be observed in Table 6.4; which is confirmed by a PCC equal to 0.890. This means a very strong positive linear correlation between them.

Agreement Metric	Sample									
	1	2	3	4	5	6	7	8	9	10
$R_{G_{AR}}$	0.691	0.688	0.684	0.578	0.767	0.541	0.617	0.525	0.604	0.619
κ_R	0.776	0.697	0.757	0.696	0.839	0.630	0.743	0.655	0.704	0.718

Table 6.4: *Agreement Metrics within Dataset.*

F1-score$_{mean}$ vs. *WiSeBE-score*

Results form Table 6.3 may give an idea that *WiSeBE-score* is just an scaled *F1-score*$_{mean}$. While it is true that they show a linear correlation, *WiSeBE-score* may produce a different system ranking than *F1-score*$_{mean}$ given the integral multi-reference principle it follows. However, what we consider the most profitable about *WiSeBE-score* is the twofold inclusion of all available references it performs. First, the construction of R_W to provide a more inclusive reference against to whom be evaluated; and then, the computation of $R_{G_{AR}}$, which scales the result depending of the agreement between all references.

6.3 Conclusion

In this chapter we presented Window-based Sentence Boundary Evaluation (WiSeBE), a semi-automatic multi-reference sentence boundary evaluation protocol based on the necessity of having a more reliable way of evaluating Sentence Boundary Detection (SBD) systems. The *WiSeBE-score* considers the performance of a candidate segmentation over a set of segmentation references and the agreement between them. It is a normalized score dependent of the performance of the candidate segmentation and the agreement between all references.

We demonstrated how *WiSeBE* is an inclusive metric which not only evaluates the performance of a system against all references, but also takes into account the agreement between them. We think this inclusivity is very important given the difficulties that are present when working with spoken language and the possible disagreements that a task like SBD could provoke.

WiSeBE-score showed to be correlated with standard SBD metrics and we hypothesize it is also correlated with extrinsic evaluations techniques like Automatic Text Summarization (ATS) and Machine Translation (MT). We base this hypothesis on the following observation: the Agreement Ratio ($R_{G_{AR}}$) of *WiSeBE-score* is strongly related to the complexity of the transcript. If a transcript is complex, annotators will have difficulties segmenting it, thus producing a low $R_{G_{AR}}$ and leading to a low *WiSeBE-score*. This same complexity affects negatively tasks like ATS and MT, meaning that a transcript with a low *WiSeBE-score* will also get low performance if ATS or MT is performed. Even this observation seems well supported, it is necessary to be validated.

Chapter 7

Transcripts Informativeness Study: an Approach Based on Automatic Summarization

Contents

Text-based Multimedia Summarization is performed with text summarization techniques without extra audio or video information in order to produce a synthetic version of the source document based on the information contained in the speech. To accomplish this, a manual or automatic transcription process is performed to obtain the corresponding text representation (Taskiran et al., 2001; Ding et al., 2012; Szaszák et al., 2016). When an Automatic Speech Recognition (ASR) system is used, it is not possible to count on a perfect transcript of the text to be summarized, thus summarization systems must be able to handle the errors produced during the transcription step. To those errors that an automatic summarization system may produce, the limitations of ASR systems are added. Therefore, we think it is essential to plan a strategy to estimate in which degree, Automatic Text Summarization (ATS) methods are influenced by transcription errors.

The performance of an ASR system is normally measured in terms of the Word Error Rate (WER) (Equation 3.10). It measures the cost of restoring the output word sequence to the original input sequence by considering the deletions, insertions and substitutions errors to computing a general error value. The lower the WER (closer to zero), the

higher its quality. ASR is nowadays the first step of further Natural Language Processing (NLP) tasks in order to solve more complex problems. Such a measure like WER seems to be effective when automatic transcription is an end by itself. Nevertheless, it does not provide any information of the negative impact the mistranscribed words may produce in further tasks (Ben Jannet et al., 2014). In this context, different measures which aim to estimate the proportion of information that is communicated have been proposed.

Relative Information Loss (RIL) (Miller, 1955), is a measure that aims to evaluate the information loss caused by ASR errors. This measure is based on mutual information to obtain the statistical dependency strength between the vocabulary of the reference and the words of the hypothesized transcript. A variation to RIL introduced by Morris et al. (2004) is Word Information Loss (WIL), which also estimates the loss of information due to transcription errors, but unlike RIL, it takes into account well transcribed words and substitutions when comparing to the reference transcript. Morris et al. (2004) also proposed Match Error Rate (MER), a variation to WER which corresponds to the probability of a given match being incorrect.

McCowan et al. (2004) suggested to take into account *Precision*, *Recall* and *F1-score* from Information Retrieval (IR) evaluation to estimate the information loss caused by drifts during the transcription process. In this framework, each word is considered an information unit and the goal of the transcription process is to retrieve all the relevant information in the original speech signal.

Ben Jannet et al. (2014) proposed to evaluate the quality of automatic transcripts in the framework of Named-entity Recognition (NER). Their method makes use of posterior probabilities to estimate the risk of error that ASR transcription errors can induce into a NER system. In a first stage, a named-entities model is built from a big collection of manually transcribed and annotated documents. Then, the model is used to compute the probability of presence of named-entities in both the automatic and manual transcripts; which are then compared to estimate the impact of transcription errors over the correct named-entities detection.

These measures have shown to be useful and to provide pertinent information for the specific tasks they were designed. Nevertheless, non of them are suitable when the goal is to perform ATS. Moreover, to our knowledge, any evaluation framework has analyze the possibility of boosting the transcription quality during evaluation. In this chapter we search to estimate the influence of the noise produced by ASR transcription errors over automatic summaries and to explore the capacity of ATS to compensate the information loss produced by transcription errors. This will be measured in terms of the informativity that is retained after the extractive summarization process.

This chapter is organized as follows. First, in Section 7.1 we explain the multilingual dataset we used to perform experiments and the steps we followed to produce automatic and reference transcripts. Then, in Section 7.2, we present the protocol we followed to evaluate the informativeness of automatic transcripts and to measure the impact of ATS over informativeness. Finally, results of the three evaluation scenarios are presented in Section 7.3.

7.1 Dataset

For a better visibility of the informativeness concept, we took into account the multilingual context through a sample of videos in English and French (10 videos per language) from the *AMIS-Dataset*. This videos cover different topics like technology, human rights, terrorism and politics in different formats like newscasts, interviews, reports and round tables. Left section of Table 7.1 displays the length of the samples within the dataset. The $\#_{en|fr}$ notation in the "Sample" column refers to the language in turn. It can be seen that the shortest sample corresponds to Sample$\#4_{en}$ with 1m42s while the longest is Sample$\#3_{fr}$ with 11m43s.

We ran three different Automatic Speech Recognition (ASR) systems to obtain the text representation of the dataset. The first two, which combine Neural Networks (NNs) and statistical models were the Google Cloud Speech API[1] (Google-ASR) and the IBM speech-to text[2] (IBM-ASR) (Saon et al., 2015). The third ASR system (AMIS-ASR) was developed by (Jouvet et al., 2018) as part of the Access Multilingual Information opinionS (AMIS) project . In addition to these three systems, we produced a manual transcript which we considered to be our gold standard in terms of transcription quality.

Central section of Table 7.1 presents the number of words each ASR system produced per sample as well as their mean and standard deviation. Gold standard transcripts are presented in the right section of Table 7.1. Analyzing the average number of words produced by each system, it can be observed that AMIS-ASR is the closest to manual transcripts in both languages, with a difference of -41.2 words for English and -8 words for French. IBM-ASR presents a difference of +65.8 words for English and -33.8 words for French with respect to manual transcripts. Finally, the difference between manual transcripts and Google-ASR in English and French is -99.8 and -110.2 words respectively. The reason for this big difference is that Google-ASR does not transcribe a word if the word confidence is lower a threshold, which means that if a sample is complex in speech terms, a lot of words are going to be excluded in the resulting transcript.

The standard deviation may give an indication of sample complexity. For those samples having a lot of noise or concurrent speakers, systems will introduce words that do not exist, or as in the case of Google-ASR, no words will be produced. This variation in number of words affects directly the standard deviation; therefore big variations of transcribed words between ASR systems will be translated into a big standard deviation.

[1] `https://cloud.google.com/speech`

[2] `https://www.ibm.com/watson/services/speech-to-ext`

Language	Sample	Length	ASR System			Mean	Manual Transcripts
			AMIS-ASR	IBM-ASR	Google-ASR		
English	1_{en}	3m19s	629	627	483	579.667 ± 68.359	621
	2_{en}	5m21s	772	784	648	734.667 ± 61.478	727
	3_{en}	2m47s	502	491	341	444.667 ± 73.441	337
	4_{en}	1m42s	262	264	219	248.333 ± 20.758	237
	5_{en}	8m47s	1 448	1 536	1 103	1 362.333 ± 186.862	1 531
	6_{en}	9m45s	1 376	1 464	1 281	1 373.667 ± 74.728	1 192
	7_{en}	5m23s	649	673	532	618 ± 61.595	644
	8_{en}	6m23s	963	988	895	948.667 ± 39.297	897
	9_{en}	7m35s	1 584	1 613	1 286	1 494.333 ± 147.789	1 595
	10_{en}	2m01s	275	266	262	267.667 ± 5.437	267
	Mean		846	870.60	705	–	804.80
French	1_{fr}	3m43s	663	585	345	531 ± 135.322	678
	2_{fr}	1m25s	239	233	225	232.333 ± 5.735	246
	3_{fr}	11m43s	2 290	2 172	2 140	2 200.667 ± 64.505	2 353
	4_{fr}	5m25s	1 009	1 009	953	990.333 ± 26.399	1 015
	5_{fr}	4m57s	894	880	837	870.333 ± 24.253	898
	6_{fr}	11m04s	2 144	2 085	1 842	2 023.667 ± 130.696	2 084
	7_{fr}	2m24s	471	467	469	469 ± 1.633	485
	8_{fr}	2m48s	483	503	461	482.333 ± 17.153	477
	9_{fr}	3m12s	572	547	529	549.333 ± 17.632	600
	10_{fr}	5m15s	731	757	673	720.333 ± 35.112	740
	Mean		949.60	923.80	847.40	–	957.60

Table 7.1: *English and French Samples from the AMIS-Dataset*

7.2 Informativeness Evaluation

It is known that the informativeness contained in a summary with respect to its source is a good indicator of the quality of the Automatic Text Summarization (ATS) system that produced the summary (Nenkova and Passonneau, 2004; Saggion et al., 2010). Therefore, we hypothesize that ATS represents an extrinsic method objective enough to evaluate a transcript produced by an Automatic Speech Recognition (ASR) system; making possible to evaluate the quality of a transcript measuring the informativeness of its corresponding summary.

In the context of ATS, the existence of sentences in the source document is essential for identifying sentences containing relevant information. However, transcripts generated by the ASR systems did not contain punctuation marks and were just a continuous sequence of words. For this reason we applied our Sentence Boundary Detection (SBD) system presented in Section 3.1.1 (CNN-B) with the English and French models over the transcripts from the three ASR systems and manual transcripts. To summarize the transcripts we opted for Autre Résumeur de TEXtes (ARTEX), an extractive ATS system described in Section 4.1, given its tolerance to noise produced by errors during the transcription phase.

Figure 7.1 exemplifies the protocol we followed to evaluate in first place the informativeness of automatic transcripts; and then, the impact of ATS over informativeness. We first obtained the manual and automatic transcripts for all samples within the dataset. Then, we applied CNN-B to segment the transcripts into Semantic Units (SUs). Next, we used ARTEX with a Compression Ratio (CR) equal to 35% to produce the cor-

responding summaries. Finally, we measured informativeness by applying different FRamework for Evaluating Summaries Automatically (FRESA) scores between manual and automatic transcripts, as well as between the manual transcripts and the automatic summaries produced by ARTEX. We proposed the three following evaluation scenarios over the FRESA scores:

- **S.1**. In this scenario FRESA is computed between manual transcripts and the automatic transcripts from the different ASR systems (AMIS-ASR, IBM-ASR, Google-ASR).

- **S.2**. During this scenario FRESA is computed between manual transcripts and the summaries produced by ARTEX. Measuring FRESA between manual transcripts and its summary establishes a maximum informativeness expected value (MaxInfEVal), which is then compared against the informativeness scores of the summaries from automatic transcripts.

- **S.3**. In this final scenario, the informativeness between S.1 and S.2 is compared to evaluate the capability of ATS to overcome informativeness loss of errors produced by ASR.

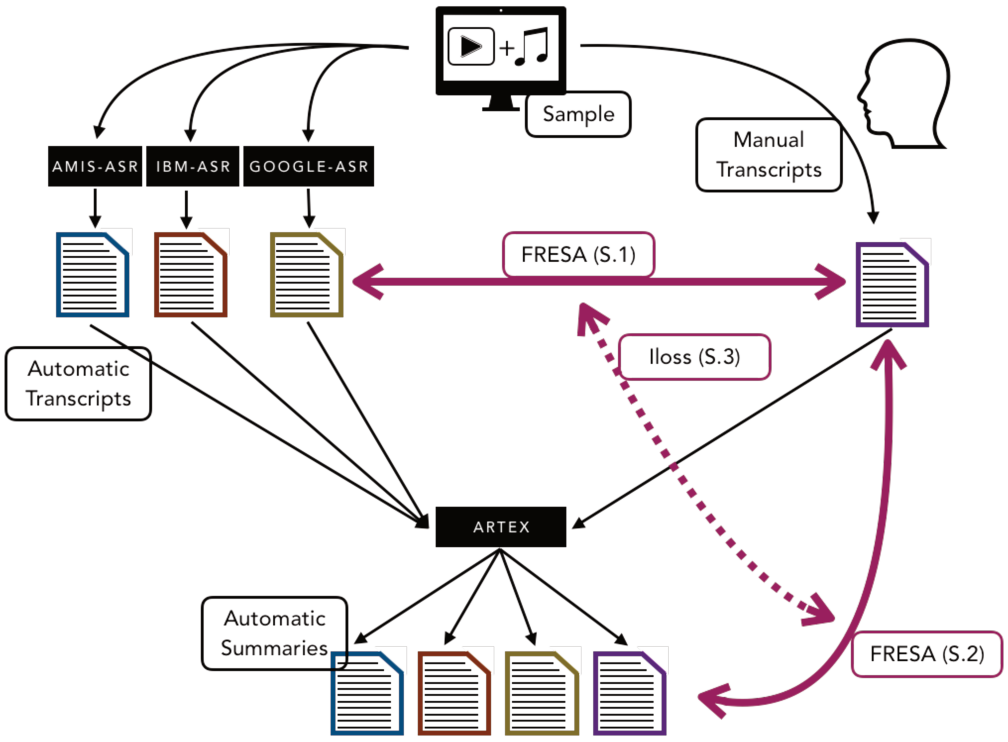

Figure 7.1: *Evaluation Protocol with Scenarios*

Language	ASR System	$\mathbf{FRESA}_1$	$\mathbf{FRESA}_2$	$\mathbf{FRESA}_4$	$\mathbf{FRESA}_M$
English	AMIS-ASR	0.741 ± 0.058	0.584 ± 0.085	0.567 ± 0.089	0.631 ± 0.076
	IBM-ASR	0.736 ± 0.055	0.578 ± 0.072	0.566 ± 0.078	0.626 ± 0.067
	Google-ASR	0.740 ± 0.088	0.605 ± 0.130	0.590 ± 0.132	0.645 ± 0.116
French	AMIS-ASR	0.835 ± 0.072	0.697 ± 0.112	0.683 ± 0.123	0.738 ± 0.101
	IBM-ASR	0.662 ± 0.134	0.485 ± 0.120	0.471 ± 0.133	0.539 ± 0.130
	Google-ASR	0.795 ± 0.125	0.664 ± 0.137	0.660 ± 0.140	0.706 ± 0.133

Table 7.2: *Manual Transcripts vs. Automatic Transcripts Results (S.1)*

7.3 Results

7.3.1 Manual Transcripts vs. Automatic Transcripts (S.1)

FRamework for Evaluating Summaries Automatically (FRESA) scores among samples for English and French are shown in Table 7.2. Google-ASR shows, in general, higher informativeness for English; while AMIS-ASR for French compared to the other Automatic Speech Recognition (ASR) systems. An interesting remark for both languages concerns the IBM-ASR system, which presents the smallest FRESA scores; but at the same time, it maintains the lowest standard deviation. This means that a low but stable informativeness level is shared over different topics and lengths. In other words; the smaller the standard deviations, the more stable the ASR system is among samples. Another remark is that for English, the difference of all FRESA scores is not statistically significant ($p > 0.05$) between the ASR systems. Nevertheless, in the case of French, AMIS-ASR and Google-ASR are statistically significant better ($p < 0.05$) that IBM-ASR.

7.3.2 Manual Transcripts vs. Automatic Summaries (S.2)

Results for the second scenario can be seen in Table 7.3. MaxInfEVal refers to the maximum informativeness expected value, which is obtained by computing FRESA between the transcript produced manually and their corresponding summaries. It corresponds to the biggest informativeness score a summary from an ASR system can obtain when measured using FRESA against manual transcripts. The closer the FRESA score to MaxInfEVal, the closer it is to be as informative as the manual transcript summary. For English and over almost all FRESA scores, AMIS-ASR shows to be more informative. For French, by contrast, Google-ASR presents the closest informativeness to MaxInfEVal. However, independently from the language, the difference of informativeness scores between ASR systems is not statistically significant ($p > 0.05$).

Language	ASR System	$\mathbf{FRESA}_1$	$\mathbf{FRESA}_2$	$\mathbf{FRESA}_4$	$\mathbf{FRESA}_M$
English	AMIS-ASR	0.395 ± 0.074	0.266 ± 0.064	0.248 ± 0.060	0.303 ± 0.063
	IBM-ASR	0.396 ± 0.075	0.261 ± 0.066	0.242 ± 0.064	0.300 ± 0.067
	Google-ASR	0.342 ± 0.081	0.222 ± 0.072	0.202 ± 0.066	0.256 ± 0.071
	MaxInfEVal	0.440 ± 0.040	0.347 ± 0.032	0.325 ± 0.031	0.371 ± 0.032
French	AMIS-ASR	0.385 ± 0.076	0.238 ± 0.064	0.213 ± 0.065	0.279 ± 0.066
	IBM-ASR	0.352 ± 0.072	0.200 ± 0.068	0.181 ± 0.065	0.244 ± 0.066
	Google-ASR	0.377 ± 0.093	0.249 ± 0.079	0.231 ± 0.082	0.286 ± 0.083
	MaxInfEVal	0.461 ± 0.061	0.371 ± 0.045	0.352 ± 0.046	0.395 ± 0.048

Table 7.3: *Manual Transcripts vs. Automatic Summaries Results (S.2)*

7.3.3 Informativeness Loss (S.3)

In this third scenario we computed the informativeness loss (*Iloss*) produced by ASR and Automatic Text Summarization (ATS) errors. For this, we first took into account the informativeness scores from both S.1 and S.2 scenarios. Then, we compared the informativeness loss of each scenario to see if ATS is capable of compensating the informativeness loss produced by transcription errors. Informativeness loss is defined as follows:

$$Iloss = 100 \times \left(1 - \frac{\text{FRESA}_{M_{\text{system-ASR}}}}{\text{FRESA}_{M_{\text{Manual Transcripts}}}}\right) \tag{7.1}$$

where $\text{FRESA}_{M_{\text{Manual Transcripts}}}$ is equal to 1 for S.1 and equal to MaxInfEVal for S.2.

Informativeness loss for all ASR systems are presented in Table 7.4. Concerning English, it can be seen that for all *Iloss* scores, information loss is smaller for **S.2** that for **S.1**. This difference is statistically significant ($p < 0.05$) for AMIS-ASR and IBM-ASR but nor for Google-ASR, which indicates that the capability of an ATS phase to reduce the effects of ASR errors depends of the ASR system that is used to perform the transcript.

Iloss scores behave different for French. IBM-ASR and Google-ASR show an informativeness loss reduction among all *Iloss* scores. That is not the case of AMIS-ASR, which information loss increases for $Iloss_2$, $Iloss_4$ and $Iloss_M$. An important remark is that for all systems, any difference between *Iloss* scores is statistically significant ($p > 0.05$).

Language	ASR System	$Iloss_1$		$Iloss_2$		$Iloss_4$		$Iloss_M$	
		S.1	**S.2**	**S.1**	**S.2**	**S.1**	**S.2**	**S.1**	**S.2**
English	AMIS-ASR	0.259	0.105	0.416	0.234	0.433	0.243	0.369	0.185
	IBM-ASR	0.264	0.106	0.422	0.250	0.435	0.260	0.374	0.196
	Google-ASR	0.260	0.227	0.395	0.364	0.410	0.383	0.355	0.316
French	AMIS-ASR	0.165	0.156	0.303	0.351	0.317	0.388	0.262	0.286
	IBM-ASR	0.338	0.221	0.515	0.460	0.529	0.483	0.461	0.373
	Google-ASR	0.205	0.173	0.336	0.326	0.341	0.340	0.294	0.270

Table 7.4: *Informativeness Loss Results (S.3)*

Language	Metric	ASR System			PCC
		AMIS-ASR	IBM-ASR	Google-ASR	
English	*WER*	0.421 ± 0.121	0.469 ± 0.109	0.414 ± 0.067	-0.472
	$FRESA_M$	0.631 ± 0.076	0.626 ± 0.067	0.645 ± 0.116	
French	*WER*	0.261 ± 0.091	0.483 ± 0.134	0.303 ± 0.131	-0.916
	$FRESA_M$	0.738 ± 0.101	0.539 ± 0.130	0.706 ± 0.133	

Table 7.5: *Manual Transcripts vs. Automatic Transcripts (WER & $FRESA_M$)*

Discussion

While Word Error Rate (WER) measures the quality of the transcript in terms of erroneous words, FRESA focalizes on the information contained within the transcript. Even this two measures focus in different aspects, it exists an intrinsic negative correlation between them. A perfect transcript will obtain a *WER*=0 and a *FRESA*=1; while a very bad one will obtain a *WER*$\approx$ 1 and a *FRESA*$\approx$ 0.

Middle section of Table 7.5 presents WER and $FRESA_M$ scores over each ASR system and language. If we compare the scores of both metrics, we can observe that while WER increases, *FRESA* decreases and vice versa. This effect is present in both language; yet, it is more evident for French. This observation is supported by the Pearson Correlation Coefficient (PCC) (Pearson, 1895) between $FRESA_M$ and WER scores for each language (right section of Table 7.5). For both cases we can observe a negative correlation produced by the inverse range interval. French presents a strong negative correlation. However, the correlation for English is not very high given that all the ASR systems show performance relatively close to 0.5 for both metrics, thus their gradients are close to zero.

7.4 Conclusion

The performance of an Automatic Speech Recognition (ASR) system is normally measured in terms of the Word Error Rate (WER). This measure takes into account deletions, insertions and substitutions errors to calculate a general value and indicate the quality of the transcript. WER is effective when transcription is an end by itself. However, when the transcript is just the first part of a NLP pipeline which final goal is to create a summary, it may more useful to get an idea of the amount of information that is contained in the transcript.

In this chapter we proposed a method to estimate the impact, in terms of informativeness, of transcription errors produced by ASR systems over further Natural Language Processing (NLP) tasks. We achieved this by computing the Kullback-Leibler (KL) divergence between the transcript generated by an ASR system and a manual transcript. We also proposed a method to indirectly reduce the informativeness loss of automatic transcripts by applying an Automatic Text Summarization (ATS) process over automatic transcripts and then comparing the informativeness loss between the automatic transcript and its summary. Results showed that in general, performing an ATS phase over automatic transcripts helps reducing the information loss produced by translation errors. Nevertheless, the significance of this improvement depends of different factors including language, ASR system and ATS method. As future work we will increase the number of samples to reduce the possible bias our small dataset may produce. Also we will vary the ATS method to analyze the impact of summarization methods over information loss.

Chapter 8

Extending Text Informativeness Measures to Passage Interestingness Evaluation

Contents

In Chapter 7 we implemented informativeness evaluation measures to study the capacity of automatic summarization to compensate the information loss produced by transcription errors. In this chapter we study the ability of state-of-the-art Focused Retrieval (FR) informativeness evaluation measures to deal with interestingness evaluation. For this, we consider short factual passages and represent them in terms of words and entities n-grams, and word embeddings. We then perform both informativeness and interestingness evaluation based on the normalized Cumulative Gain (nCG) for different passages cut-off values.

FR is an extension of document retrieval where it is not only important to recover a set of relevant documents, but to locate the informative passages inside those documents; thus providing the user with a direct access to the desired information (Kamps et al., 2008). This information can be factual and explicitly linked to the query as in Question Answering (QA) or more abstract and provide some general background about the query. FR systems are evaluated according to their informativeness (Bellot et al., 2016), which corresponds to the cumulative length of the extracted informative

passages and their . Given a query representing a user information need, a FR system returns a ranked list of short passages extracted from a document collection. Then, a user reads a passage top to bottom and tag it as: 1) informative, 2) partially informative or 3) uninformative depending if all, only parts or no part of it contains useful information relevant to the query.

Interestingness, by contrast, refers to a much broader concept used in Data Mining (DM) as it is defined as the power of attracting or holding attention. It relates the ideas of lift and information gain used to mine very large sets of association rules between numerous features which is a complex interactive process; and unlike informativeness, there is no precise query to initiate the search. For the purpose of this study, we define interestingness in the context of FR as: *a text passage that is clearly informative for some implicit user's information need.* More precisely; given a set of users and a set of passages that were considered interesting by at least one of these users, the task consists in finding new interesting passages not related to previous topics and explicit queries.

This Chapter is organized as follows. In Section 8.1 we explain our methodology to extend text informativeness measures to passage interestingness evaluation. We first explain the dataset we used to perform all evaluations, followed by the process we adopted to evaluate informativeness and interestingness. Then in Section 8.2 we present the obtained results over both informativeness and interestingness over two experimental scenarios.

8.1 Experimental Setup

We conceptualize short passage informativeness evaluation as a ternary relation (Figure 8.1 (A)) between a set of topics (T), a set of short passages (P) from a large collection and a set of graded scores (S) such that top ranked passages contain certainly relevant information about the related topic or its background. Given this ternary relation it is possible to define short passage interestingness evaluation as a projection of informativeness over P and S, which produces a binary relation (Figure 8.1 (B)) between a set of short passages and graded scores for some unknown topic.

8.1.1 Dataset

In this chapter we aim to analyse the ability of state-of-the-art informativeness evaluation measures to deal with interestingness evaluation. For this kind of study it is necessary to count with a big controlled dataset with the particularity of having clear and defined topics with an informativeness score associated to each one of them. The *AMIS-Dataset*, as described in Section 1.2.1, covers different topics that are well defined. However, the transcripts associated to each one of the samples have been obtained automatically via an Automatic Speech Recognition (ASR) system. Added to this, the *AMIS-Dataset* unfortunately does not count with informativeness scores assigned to each the samples, invalidating its possibility to be used in this study.

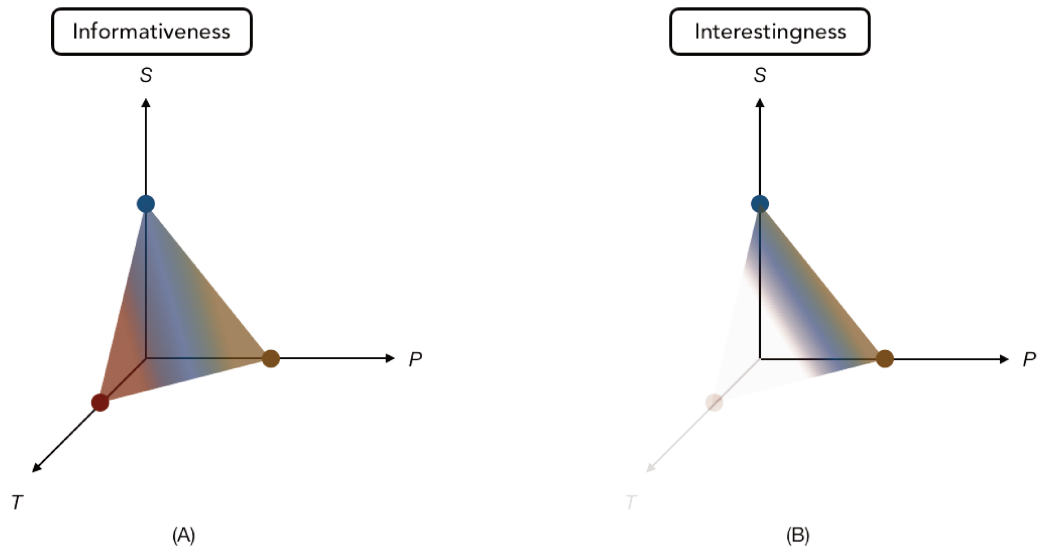

Figure 8.1: *Informativeness and Interestingness Projection (T: Set of Topics; P: Set of Short Passages; S: Set of Graded Scores)*

We opted for the data collection from the Tweet Contextualization Track at INEX 2012 (TC@INEX 2012) task (SanJuan et al., 2012). The participants to this task had to provide a summary composed of extracted passages from Wikipedia[1] that should contextualize a tweet by revealing its implicit background and providing some factual explanations about related concepts. All tweets were collected from non-personal Twitter[2] accounts.

The TC@INEX 2012 data collection provides the three elements needed during short passage informativeness evaluation (Figure 8.1 (A)):

- A set of 63 different topics (T). A topic is a short sentence (a tweet) that was used by participants to build a query-driven summary.
- A set of 671 191 passages from 33 valid runs (P): A run consists of several summaries, one per topic, which had to be built by participants. Each summary is 500 words-long or less and is formed by passages (sentences) from Wikipedia.
- A set of 34 519 graded passages (S): Passages from a set of summaries were marked as informative, partially informative or non informative by human assessors.

Each topic was evaluated by two people; therefore each graded passage has a score among $[0, 1] \cup \{2\}$ depending of the relative length that was highlighted as informative by at least one evaluator. In the case of agreement by both evaluators that the whole passage was informative, an score of two was assigned. Human scores can be seen in Table 8.1, most passages were marked as non informative by both evaluators while only

[1] `https://www.wikipedia.org/`
[2] `https://twitter.com`

Informative Score Ranges	# of Passages	Percentage
2	306	0.886%
[1,2)	5 460	15.817%
(0,1)	768	2.225%
0	27 985	81.071%
Total	34 519	100%

Table 8.1: *Human Evaluation Passages Scores*

19% was considered at least as partially informative by one evaluator. If all passages that were selected as informative are considered and separated by topic, it is possible to build a textual reference (φ_τ) for each topic to evaluate informativeness. By contrast, by merging all topics, it is possible to obtain a textual reference (δ_τ) to evaluate interestingness.

From each passage we extracted different discrete and continuous text units. Discrete text units correspond to uni-grams, bi-grams and skip-grams of word stems which we obtained with the Porter Stemmer algorithm (Van Rijsbergen et al., 1980) after a process of stop words removal. Added to this discrete units, we also considered uni-grams, bi-grams and skip-grams of Wikipedia entities in anchor texts. Regarding continuous text units, we created two different word2vec models, each one composed of 300 dimensions and negative sampling of 15. The google_news$_{1gram}$ model, composed of 3 million embeddings was the same used by Ng and Abrecht (2015) to calculate ROUGE-WE. The clef_inex$_{1gram}$ model, composed of 30 thousand embedding, was trained with all passages of the TC@INEX 2012 data collection.

8.1.2 Text Informativeness Evaluation Measures

We evaluated the following textual overlap measures:

- F_{1_1} : F1-score among uni-grams
- F_{1_2} : F1-score among bi-grams
- $F_{1_{sk}}$: F1-score among skip-grams with a gap of one word
- KL_1 : KL divergence among uni-grams
- KL_2 : KL divergence among bi-grams
- KL_{sk} : KL divergence among skip-grams with a gap of one word
- $LogSim_1$: LogSim score among uni-grams
- $LogSim_2$: LogSim score among bi-grams
- $LogSim_{sk}$: LogSim score among skip-grams with a gap of one word
- $w2v_{g_1}$: Word2vec cosine similarity over google_news$_{1gram}$
- $w2v_{c_1}$: Word2vec cosine similarity over clef_inex$_{1gram}$

The first nine correspond to discrete overlap measures based on smoothed probabilities (except from F1-scores) over all text units. Nevertheless, we also considered useful to include only a relevant subset of the text units in form of nuggets.

Precision, Recall & F_β-score

Text informativeness measures in the context of FR are defined in terms of the type of text unit Ω (words, lemmas, stems, etc.) from a textual reference R and a sentence S which informativeness is to be evaluated. *Precision* corresponds to the intersection between text units of S and R divided by the number text units of S:

$$Precision = \frac{|\Omega(S) \cap \Omega(R)|}{|\Omega(S)|} \tag{8.1}$$

Recall corresponds to the intersection between text units of S and R divided by the number text units of R:

$$Recall = \frac{|\Omega(S) \cap \Omega(R)|}{|\Omega(R)|} \tag{8.2}$$

The F_β-score corresponds to the harmonic mean between *Precision* and *Recall*. It is defined as:

$$F_\beta = \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{8.3}$$

where β is the factor that controls the relative emphasis between *Precision* and *Recall*. If $\beta = 1$ neither *precision* nor *recall* is favored, simplifying Equation 8.3:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{8.4}$$

F1-score (F_1) is the most common normalized set theoretic similarity giving equal emphasis to *precision* and *recall*. To represent F_1 in terms of Ω, R and S, it is just necessary to substitute equations 8.1 and 8.2 in Equation 8.4:

$$F_1 = 2 \times \frac{|\Omega(S) \cap \Omega(R)|}{|\Omega(S)| + |\Omega(R)|} \tag{8.5}$$

Kullback-Leibler (KL) Divergence

For cases where a very large number of documents have to be evaluated, reference summaries availability becomes a major issue. Thus, it becomes easier to apply a measure

that can be used automatically to compare the content of the candidate summaries with the full set of source documents rather than comparing with human created reference summaries. In this framework, measures such as Kullback-Leibler (KL) probability distribution divergence used in Text Analysis Conference (TAC) (Dang and Owczarzak, 2008) compare the probability distributions of text units between S and R. In this approach, informativeness relies on complex hided word distributions and not on specific units.

We implemented KL divergence D_{KL} as the expectation based on the reference R of the logarithmic difference between normalized frequencies in R and smoothed probabilities over S. This divergence is not normalized.

$$D_{KL}(R||S) = \sum_{\omega \in \Omega(R)} \ln \left(\frac{P(\omega|R) \cdot (|S|+1)}{P(\omega|S) \cdot |S| + P(\omega|\Omega)} \right) \cdot P(\omega|R) \tag{8.6}$$

where $|R|$ and $|S|$ correspond to the cardinality of R and S respectively.

Logarithm Similarity (LogSim)

The TC@INEX 2012 (SanJuan et al., 2012) provided their participants with the Logarithm Similarity (LogSim) measure, a normalized ad-hoc dissimilarity able to sort three main issues:

- Nonexistence of human references.
- Variability on the size of automatic summaries.
- Existence of very short automatic summaries.

Like the KL divergence, LogSim is also an expectation based on the reference R but of a normalized similarity that is only defined over $\Omega(S) \cap \Omega(R)$.

$$LogSim(S||R) = \sum_{\omega \in \Omega(S) \cap \Omega(R)} e^{-\left|\ln\left(\frac{L_R(\omega,S)}{L_R(\omega,R)}\right)\right|} \times P(\omega|R) \tag{8.7}$$

where,

$$L_R(\omega, \Theta) = \ln(1 + P(\omega|\Theta) \times |R|) \tag{8.8}$$

Nugget-based Evaluation

It is likely that automatic summaries that contain highly frequent terms are less interesting for a user than a summary that contains less frequent and thus probably more informative terms. In the same way, it is likely that summaries that contain Wikipedia entities are more informative than summaries that do not contain any. For this, we considered the anchors associated with Wikipedia entities as potential nuggets as defined in Pyramid evaluation (Dang, 2005).

In the context of QA, Dang et al. (2007) defined a nugget as an Informative Text Unit (ITU) about the target that is interesting; atomicity being linked to the fact that a binary decision can be made on the relevance of the nugget to answer a question. This method makes it possible to consider documents that have not been evaluated to be labeled as relevant or not relevant (simply because they contain relevant nuggets or not). It has been shown that real ITUs can be automatically extracted to convert textual references into a set of nuggets (Ekstrand-Abueg et al., 2013). This simplifies the problem of informativeness evaluation, providing a method to measure the proximity between two sets of ITUs.

The general idea is that informativeness relies on the presence or absence of some specific text units (nuggets) and can be then evaluated based on their counting. If nuggets are unambiguous entities, standard discrete measures based on smoothed probabilities or Pyramid measures can be used (Lin and Zhang, 2007) and more sophisticated nugget score measures based on shingles if not (Pavlu et al., 2012). The corresponding nugget based measures we analyze are: $F_{1_{1-ent}}$, $F_{1_{2-ent}}$, $F_{1_{sk-ent}}$, $LogSim_{1-ent}$, $LogSim_{2-ent}$ and $LogSim_{sk-ent}$.

The remaining two measures correspond to cosine similarities over continuous space word representations. As proposed by Mikolov et al. (2013b), it is possible to combine words by an element-wise addition of their embeddings. Given a document D of length n represented by the set of word embeddings $D = \{\overline{v}_1, \overline{v}_2, ..., \overline{v}_n\}$ where $|\overline{v}_i| = m$; a simple way to represent D with a unique embedding in terms of D is to sum each component j of each embedding $\overline{v}_i$ in D to obtain a unique vector $\overline{d}$ of length m. Then, to measure the similarity between the vector of a reference document ($\overline{d}_R$) and the one of a proposed sentence ($\overline{d}_S$), the cosine similarity between the two vectors is calculated as:

$$cos_{RS}(\theta) = \frac{\overline{d}_R \cdot \overline{d}_S}{|\overline{d}_R| \cdot |\overline{d}_S|} \tag{8.9}$$

8.1.3 Informativeness and Interestingness Evaluation

To evaluate informativeness, for each measure $\mu \in \mathcal{M}$ and passage $\omega \in \Omega$ associated to a topic $\tau \in \mathcal{T}$, we computed $\mu(\varphi_\tau, \omega)$ to estimate the overlap between ω and the reference φ_τ defined as:

$$\varphi_\tau \quad = \quad \bigcup \{\omega \in \Omega : \tau \in \mathcal{T}, ref_\tau(\omega) > 0\} \tag{8.10}$$

where $ref_\tau(\omega)$ corresponds to the manually assigned score of ω in the topic τ.

On the other hand, to evaluate interestingness, for each measure $\mu \in \mathcal{M}$ and passage $\omega \in \Omega$ associated with a topic $\tau \in \mathcal{T}$, we computed $\mu(\delta_\tau, \omega)$ to estimate the

overlap between ω and the reference δ_τ, defined as the concatenation of passages that are informative for at least one different topic $\tau' \neq \tau$:

$$\delta_\tau = \bigcup \{\omega \in \Omega : (\exists \tau' \in \mathcal{T} - \{\tau\}), ref_{\tau'}(\omega) > 0\} \tag{8.11}$$

where $ref_{\tau'}(\omega)$ corresponds to the manually assigned score of ω in the topic τ'. In order to avoid any overfitting effect, we split the dataset ranked per topic into 10 folds and restricted δ_τ to passages in a different fold than the one including τ.

Lin and Hovy (2003) evaluated the effectiveness of automatic evaluation measures by calculating the correlation between systems' average measure scores and their human assigned average scores; however, we adopted a different procedure. We first took into account each passage individually and computed all scores for each measure $\mu \in \mathcal{M}$. Then, to evaluate one specific measure, we followed the same approach as in Yilmaz et al. (2015) and ranked in decreasing order the graded passages by the measure. Finally, considering different passage cut-off values, we computed the nCG over the top ranked passages. The nCG of a measure at a cut-off value corresponds to the sum of the graded human judgments top ranked passages (by the measure) divided by the maximum score that could have been expected at a precise cut-off value.

Given a set of topics $\mathcal{T}$ and a set of passages Ω for which there is a graded evaluation $ref_\tau(\omega)$ of their informativeness for at least one topic $\tau \in \mathcal{T}$, $nCG_k(\mu)$ was computed as follows for any measure $\mu \in \mathcal{M}$ and any cut-off value k:

$$nCG_k(\mu) = \frac{\max\{\sum_{\omega \in S} \mu(\Theta, \omega); S \subset \Omega, |S| \leq k\}}{\max\{\sum_{\omega \in S} ref_\tau(\omega); S \subset \Omega, |S| \leq k, \tau \in \mathcal{T}\}} \tag{8.12}$$

where $|S|$ is the cardinal of S and $\Theta = \{\varphi_\tau, \delta_\tau\}$.

If the cut-off value k is lower than the number of passages S such that $ref_\tau(\omega) > 0$ where $\omega \in S$, $nCG_k(\mu)$ reflects a precision value. On the contrary, for a cut-off value k higher than the number of passages S such that $ref_\tau(\omega) = 0$ where $\omega \in S$, $nCG_k(\mu)$ indicates the maximal recall that can be expected using this measure.

8.2 Results

We conducted two experimental scenarios to evaluate the effectiveness of each measure $\mu \in \mathcal{M}$. In the first scenario (SC.A) we focused on informativeness to test which measure ranked first most informative passages per topic. During the second scenario (SC.B) we focused on interestingness to test which measure ranked first most interesting passages independently from the topic. For both scenarios we computed the $nCG_k(\mu)$ of any measure μ at the following cut-off values k =100, 500, 1 000, 2 500, 5 000, 10 000. Cut-off values bigger than 6 543 allow to analyze the maximal recall a measure may achieve.

8.2.1 SC.A: Informativeness Evaluation

Discrete Overlap Measures

Performance of discrete overlap measures is shown in Figure 8.2. It can be seen that all *KL* measures performed similar over all cut-off values, with the lowest *nCG* scores. Among all measures over uni-grams, F_{1_1} achieved the highest score for all cut-off values; obtaining a maximal precision of 0.45 for k=2 500 and a maximal recall of 0.58 for k=10 000. F1-scores over bi-grams (F_{1_2}) and skip-grams ($F_{1_{sk}}$) overperformed LogSim over the same units ($LogSim_2$, $LogSim_{sk}$); however this improvement is not statistically significant ($p > 0.05$). $F_{1_{sk}}$ reached the highest precision of 0.77 for k=5 000. Maximal recalls over 0.92 were reached only bi-gram based measures (F_{1_2}, $F_{1_{sk}}$, $LogSim_2$, $LogSim_{sk}$) for k=10 000.

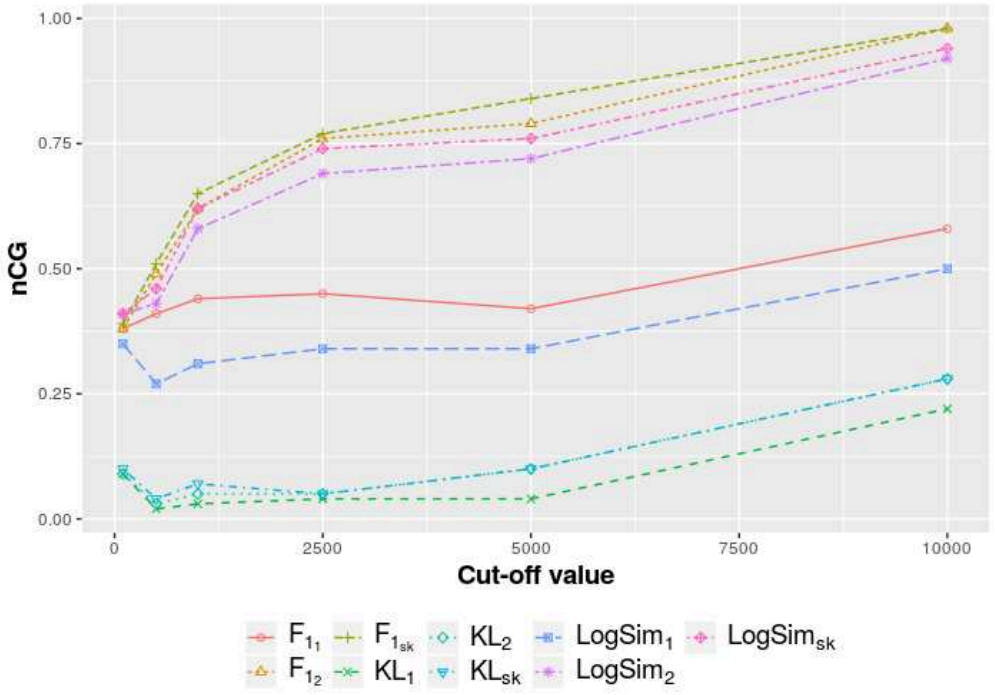

Figure 8.2: *nCG Scores Over Discrete Overlap Measures (Informativeness)*

Discrete Overlap Measures Restricted to Wikipedia Entities

In Figure 8.3 we can observe the performance of discrete overlap measures restricted to Wikipedia entities. Compared to F1-scores restricted to entities, all LogSim measures ($LogSim_{1-ent}$, $LogSim_{2-ent}$, $LogSim_{sk-ent}$) had very low *nCG* scores for cut-off values $k \leq$ 2 500. Nevertheless, $LogSim_{sk-ent}$ achieved the highest precision over all restricted measures for k=5 000 followed by $LogSim_{2-ent}$. For k=10 000, $LogSim_{2-ent}$ obtained the highest recall followed by $LogSim_{sk-ent}$. $F_{1_{2-ent}}$ and $F_{1_{sk-ent}}$ presented a similar behavior

over all cut-off values with the particularity that $F1_{sk-ent}$ had a slightly higher precision and $F1_{2-ent}$ a slightly higher maximal recall. Except for $F1_{1-ent}$, all measures present a lower performance compared to their counterparts measures with standard word n-grams.

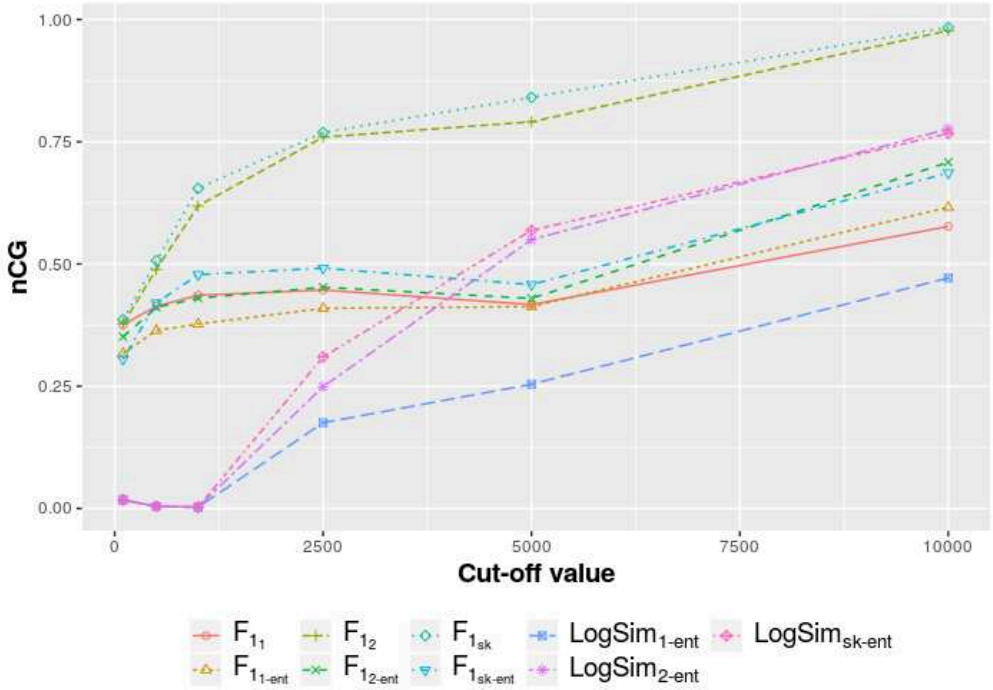

Figure 8.3: *nCG Scores Over Wikipedia Entities (Informativeness)*

Discrete vs. Continuous Measures

The performance of $w2v_{c_1}$ and $w2v_{g_1}$ continuous measures is presented in Figure 8.4. Continuous measures seemed to perform poorly compared to discrete F1-scores over uni-grams, bi-grams and skip-grams. Both measures behaved similar performance for all cut-off values; however, $w2v_{c_1}$ slightly overperformed $w2v_{g_1}$. It reached a maximal precision of 0.32 for k= 5 000 and a maximal recall of 0.46. This behavior can be explained by the fact that the amount of unknown words in the $clef\_inex_{1gram}$ model is smaller that the one in the $google\_news_{1gram}$ model.

8.2.2 SC.B: Interestingness Evaluation

Discrete Overlap Measures

Figure 8.5 shows the performance of discrete overlap measures for interestingness. Except for $F1_1$ and $LogSim_1$, all measures obtained their maximal precision for k=2 500. All

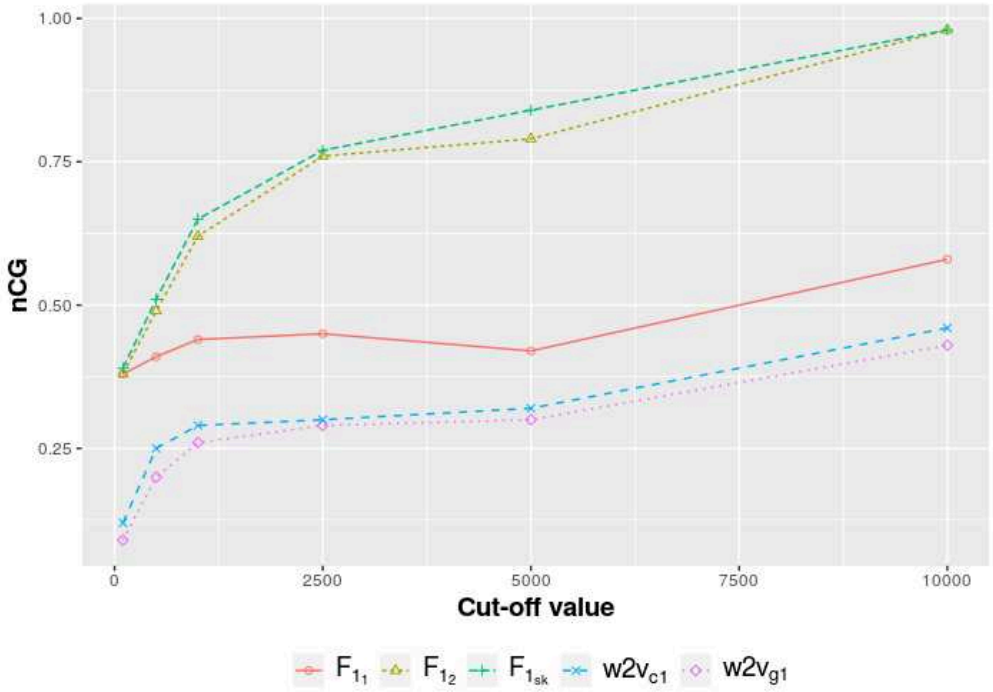

Figure 8.4: *nCG Scores Over Continuous Measures (Informativeness)*

KL measures converged at k=5 000 with a precision of 0.32 and achieved a similar maximal recall for k=10 000. Similar to informativeness results (Figure 8.2), highest *nCG* scores were obtained by F1-scores and LogSim measures for bi-grams and skip-grams. F1-score over bi-grams (F_{1_2}) and skip-grams ($F_{1_{sk}}$) behaved very similar for all cut-off values; both achieved a maximal precision of 0.75. $LogSim_2$ and $LogSim_{sk}$ obtained a perfect recall for k=10 000, while F_{1_2} and $F_{1_{sk}}$ arrived to 0.7. F_{1_1} and $LogSim_1$ presented the lower performance over all cut-off values. They both achieved a maximal precision of 0.07 for k=5 000 and a maximal recall of 0.23.

Discrete Overlap Measures Restricted to Wikipedia Entities

The impact of restricting references and passages to Wikipedia entities in the case of interestingness can be seen in Figure 8.6. Lowest scores for all cut-off values were obtained by uni-gram based measures $F_{1_{1-ent}}$ and $LogSim_{1-ent}$. $F_{1_{2-ent}}$ achieved a maximal precision of 0.47 for k=5 000 and a similar maximal recall than its counterpart measure with standard word n-grams. Best *nCG* scores for all cut-off values were obtained by $LogSim_{2-ent}$ and $LogSim_{sk-ent}$. Both achieved a maximal precision of 0.92 for k=2 500 and a maximal recall of 0.94. Similar to informativeness results, all measures presented a lower performance compared to their counterparts measures with standard word n-grams.

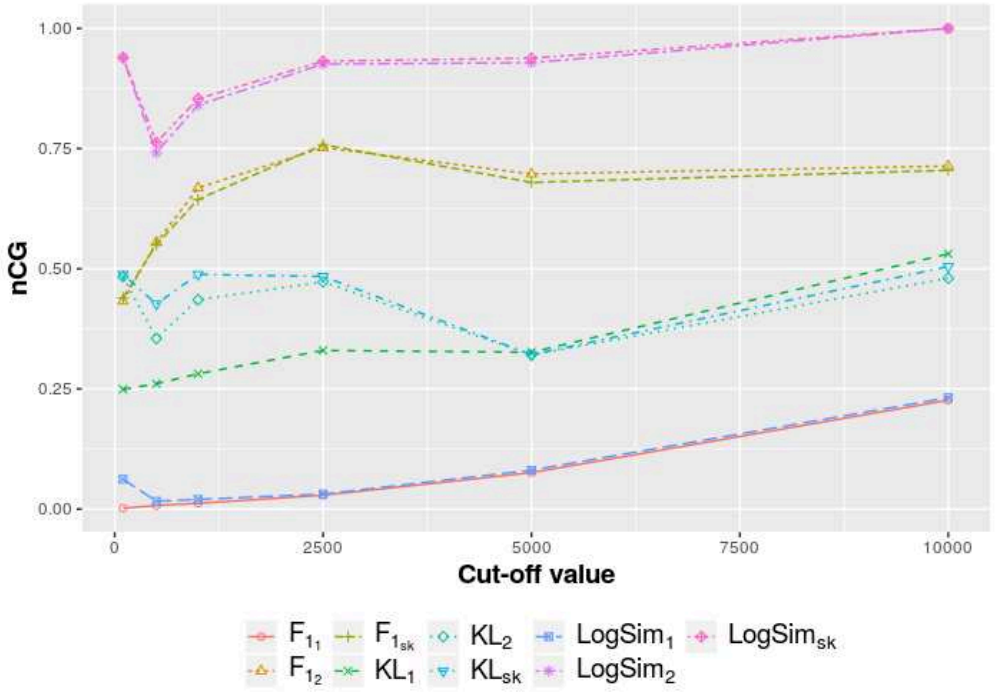

Figure 8.5: *nCG Scores Over Discrete Overlap Measures (Interestingness)*

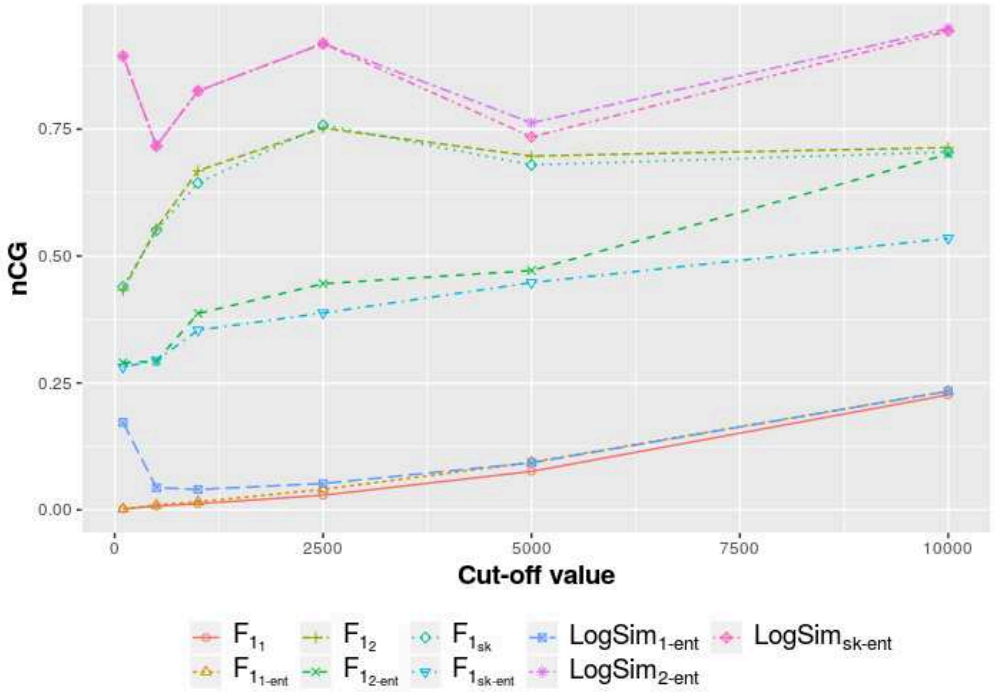

Figure 8.6: *nCG Scores over Wikipedia Entities (Interestingness)*

Discrete vs. Continuous Measures

Figure 8.7 presents the results of uni-gram ($w2v_{g_1}$, $w2v_{c_1}$) continuous measures and compares them against F1-scores discrete metrics over uni-grams (F_{1_1}), bi-grams (F_{1_2}) and skip-grams ($F_{1_{sk}}$). It can be seen that for all cut-off values F_{1_2} and $F_{1_{sk}}$ overperformed all continuous measures. For low cut-off values they arrived to a maximal *precision* of around 0.75, while for high cut-off values they achieved a maximal recall of 0.7. Continuous measures showed a similar behaviour over all cut-off values; however, $w2v_{g_1}$ was slightly better with a maximal precision of 0.2 for k=5 000 and a maximal recall of 0.36.

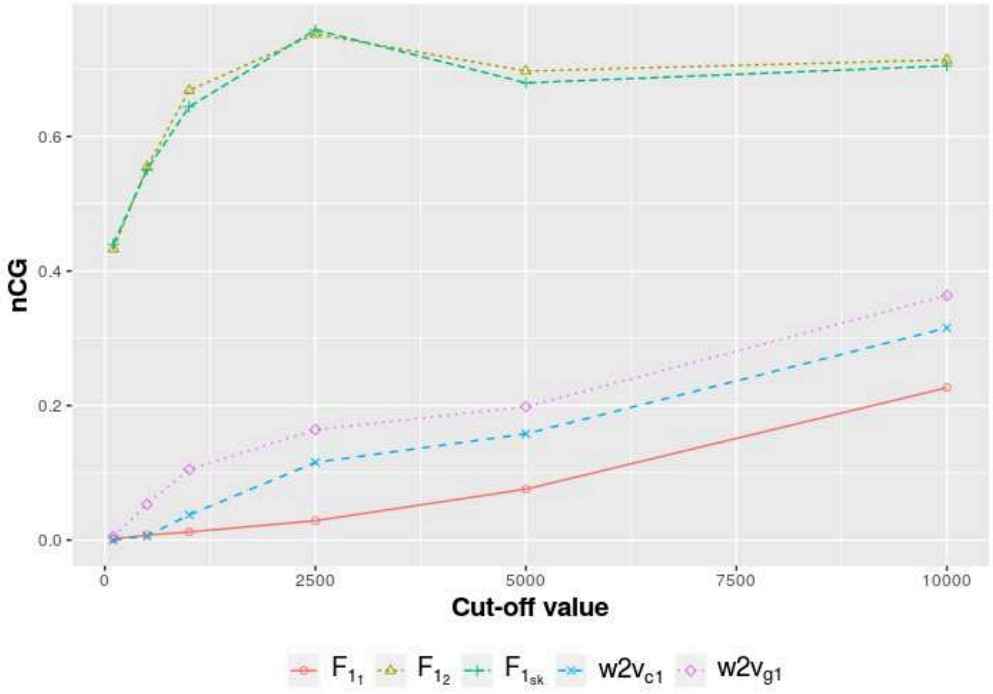

Figure 8.7: *nCG Scores Over Continuous Measures (Interestingness)*

8.3 Conclusion

In this chapter we defined the concept of interestingness in the context of FR as a generalization of the concept of informativeness. From each passage within the TC@INEX 2012 dataset we extracted different discrete and continuous text units. Discrete text units corresponded to uni-grams, bi-grams and skip-grams of word stems and Wikipedia entities in anchors texts (nuggets); while continuous text units corresponded to word2vec uni-gram embeddings. We then studied the ability of state-of-the-art short passage informativeness evaluation measures to deal with interestingness evaluation. For this, we performed both informativeness and interestingness evaluation of each measure based

on its nCG for different cut-off values.

Based on the results explained in Section 8.2 we can conclude the following:

- F1-scores over bi-grams and skip-grams of words are the most suitable measures for informativeness evaluation.
- LogSim measures over bi-grams and skip-grams of words are the most suitable measures for interestingness evaluation.
- All measures over uni-grams of words have general low performance; yet this behaviour is more evident for interestingness evaluation.
- Nugget evaluation over bi-grams and skip-grams decrease performance for informativeness and interestingness evaluation.
- KL measures perform better over interestingness evaluation compared to informativeness evaluation.

Informativeness measures showed to be useful for both informativeness and interesting evaluations, which opens the possibility of exploring further evaluation measures in the context of Automatic Text Summarization (ATS).

Chapter 9

Conclusions and Perspectives

Contents

9.1 Conclusions

Online availability have made of multimedia sources an easy access information resource, which has raised the need of helping the understanding of the large amount of information they produce. One way to approach this demand is by summarizing the multimedia content, thus generating an abridge and informative version of the original source. In this PhD thesis we addressed the subject of text and audio-based multimedia summarization in a multilingual context. It was developed in the frame of the AMIS CHISTERA-ANR project, which intends to make information easy to understand to everybody.

The first part of this PhD thesis was dedicated to text and audio-based multimedia summarization with a multilingual perspective. In a multilingual context, a document to be summarize can be an audio signal. If the summarization process is wanted to be driven by textual methods, a transcription process, normally with an Automatic Speech Recognition system, has to be performed. A major issue of transcripts produced by these systems is their lack of syntactic information; specially sentence separation. Sentence Boundary Detection aims to restore part of the syntactic information separating the transcript into sentences. We approached this task as a binary classification problem with subword-level information vectors and Convolutional Neural Networks to predict if the central word inside a sliding context window corresponded or not to a sentence boundary. Results of the different experiments we performed showed that our approach have a similar performance for French and English. However, it presents certain difficulties processing Arabic given its morphological complexity and the way

sentences are formed. Concerning cross-domain evaluation, tuning our segmentation model with a in-domain dataset showed to have a positive impact concerning its recall; however, it had a repercussion in precision. The unbalanced nature of the data is a latent problem that affects the segmentation task and our approach was not the exception.

After speech recognition and sentence segmentation processes have been applied over the audio signal, multimedia summarization can be driven by text summarization methods. We created ARTEX-MSA, a new text-based extractive summarizer for Arabic based on ARTEX. Summarizing Arabic transcripts is challenging given the complexity of the language in terms of morphology and structure; added to the domino effect errors produced by the concatenation of speech recognition and sentence segmentation. ARTEX-MSA has the attribute of not needing a training corpus, which is an advantage given the difficulties regarding public summarization Arabic resources. Comparative results showed that ARTEX-MSA has an average performance, yet it benefits of being lightweight, portable and language scalable. ARTEX-MSA has the limitation of creating a general summaries, which is a problem for transcripts where multiple topics are discussed and is desirable to recover pertinent segments of each different topic.

Multimedia summarization of an audio document can also be performed by directing the summary using only audio features. We developed an audio-based multimedia extractive summarizer that during training phase models the informativeness of a set of transcripts in terms of their audio features. Then, when a summary is to be created, no text representation is required and the informativeness of the audio segments is derived from the trained model. Introducing the informativeness in audio-based summarization is an original and powerful contribution which has the capacity of producing an audio summary that is comprehensible, agreeable and informative. Even if a training phase is required, the only resource that is needed is the text representation of the source audio documents. Manual evaluation showed that summaries generated by this method are in average half informative. However, when summary segments were evaluated independently, the informativeness level decreased given the short length of each segment. The method also manages to select audio segments all along the source document, which is good when the information is well distributed all over the source audio document.

The second part of this PhD thesis was dedicated to evaluation measures and the subjectivity of gold standards in certain Natural Language Processing tasks. We proposed WiSeBE, a new semi-automatic multi-reference sentence boundary evaluation protocol that addresses the following questions: Given a transcript, does it exist a unique segmentation reference? Or, is it possible that the same transcript could be segmented in five different ways by five different people in the same conditions? If so, which one is correct? And more important, how to fairly evaluate the automatically segmented transcript? We showed how WiSeBE not only evaluates the performance of a system against all references, but also takes into account the (dis)agreement between them. This is fundamental when working with spoken language given the subjectivity in gold standards that a task like Sentence Boundary Detection causes. WiSeBE showed to be correlated with other agreement measures, validating its capacity of op-

erating from a multi-reference perspective. Notwithstanding the great advantages a multi-reference sentence boundary evaluation protocols like WiSeBE provides, multi-reference availability is expensive and is not always an option. In cases when only one reference is available, it is always possible to implement WiSeBE to profit from its window-based evaluation approach.

Performance of speech recognition systems is normally measured in terms of the word error rate. This measure is effective when the transcript is an end by itself. However, when the final objective is to create a summary and the transcript is just the first part of the pipeline, it may be more useful to have a measure that reflects the amount of information that is contained in the transcript. We proposed an original methodology to measure the quality of an automatic transcript in terms of its informativeness; and in which grade automatic summarization may compensate the information loss raised during transcription phase. This methodology uses an automatic text summary evaluation protocol without reference, which computes the divergence between probability distributions of different textual representations: manual and automatic transcripts, and their summaries. Results over English and French showed that in general, performing a summarization phase over automatic transcripts helps reducing the information loss produced by translation errors. Nevertheless, the significance of this improvement depends of factors like language, speech recognition system and summarization method.

Finally, we presented a study on how discrete and continuous text informativeness measures may be extended to passage interestingness evaluation. We performed both informativeness and interestingness evaluation with a set of informativeness measures over a big controlled dataset. Results showed interesting similarities between informativeness and interestingness evaluation; both seemed to be better represented with bi-grams and skip-grams of words than with simple uni-grams. Also, using word unigrams and nuggets had a negative impact over all measures for both cases; however, it was more evident for interestingness. LogSim seemed to be the best informativeness measure to interestingness evaluation; however, it must be used with bi-grams or skip-grams of words. These observations provide useful insight regarding interestingness evaluation and open the possibility of implementing other measures in the framework of automatic summarization.

9.2 Perspectives

During the development of this PhD thesis we addressed a variety of topics related to multimedia summarization with a multilingual perspective. This has lead to various contributions that enrich the state-of-the-art and hopefully are useful to people outside the academic domain. However all our contributions are far from perfect and can be improved, adapted and extended. We separate our perspectives in three main axes: Sentence Boundary Detection, multimedia summarization and informativeness evaluation.

Concerning Sentence Boundary Detection, we would like to extend our solution into a hybrid system capable of processing audio and textual features in a voting scheme. We would consider the speaking speed rate of each speaker utterance to obtain a first indicator of possible sentence boundaries. Then, we would apply our existing segmentation system over those latent sentence boundaries to validate or reject the boundary. Sentences in Arabic are usually connected by particles and not by punctuation marks. The most common way is by the letter wa (و). In addition to the hybrid system, we would like to include a language dependent module to consider language particularities like the one mentioned before. Lastly, we would also conduct an study to measure the correlation between WiSeBE and extrinsic evaluation techniques like automatic text summarization and machine translation.

With respect to text-based multimedia summarization, we would like to provide a statistical story segmentation method to detect topic changes between sentences with no constraint on the number of different topics. We plan to consider lexical similarities between sentences inside a sliding window to detect thematic changes. Regarding audio-based multimedia summarization, we would like to experiment with different scoring functions over our informativeness based summarizer approach in order to include information like position and length; always giving priority to informativeness.

In the case of informativeness, we would like to apply our transcripts evaluation methodology over a bigger dataset and considering other languages to validate the obtained results. Finally, we would like to experiment with other informativeness measures over interestingness evaluation. For this we would try the framework of automatic text summarization and measures like ROUGE, FRESA and SummTriver (Torres-Moreno, 2015; Cabrera-Diego and Torres-Moreno, 2018).

We plan to create public repositories with GNU General Public License (GPL) of our contributions. We find this to be the best way of sharing and giving value to our research.

Appendix A

Data Visualisation

In Figure A.1 it can be observed the number of Semantic Units (SUs) for each one of the 30 samples from the *AMIS-Dataset30*. In average, transcripts are constituted of 43 ± 88 SUs. The shortest sample is Sample#7 with only 2 SUs and the longest one is Sample#20 with 395 SUs.

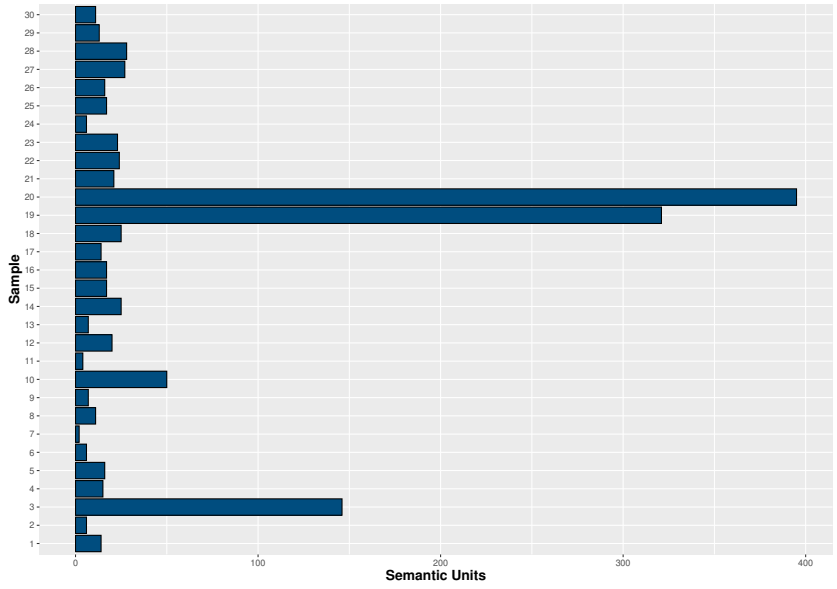

Figure A.1: *Semantic Units per Sample in AMIS-Dataset30*

Figure A.2 displays the SUs average length for each sample in *AMIS-Dataset30*. In average, the number of words of each SU for all transcripts is 13 ± 4. Sample#10 contains the shorter SUs in average with 8 words per SU, being Sample#7 the one that contain the longest SUs in average with 26 words per SU.

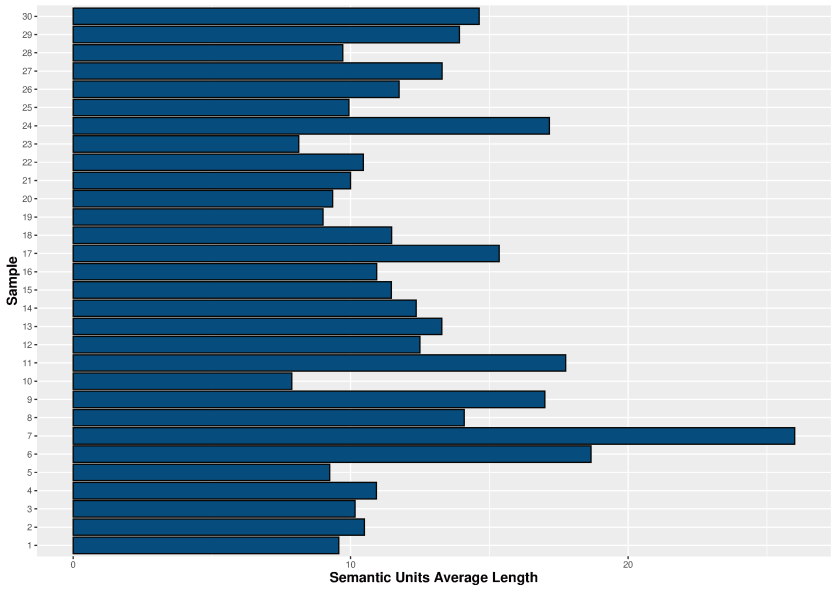

Figure A.2: *Average Semantic Unit Length per Sample in AMIS-Dataset30*

Results regarding the Audio Features Summarizer described in Section 5.2 are shown from Figure A.3 to A.12. Blue rectangles correspond to each one of the segments of a sample. Score of each segment is represented by its height.

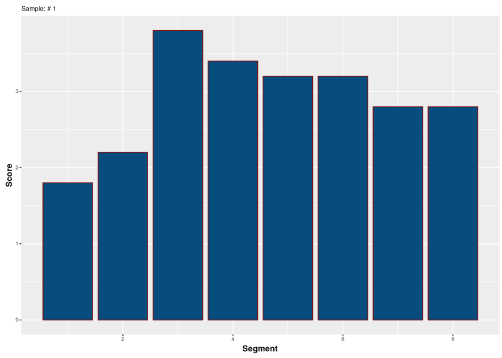

Figure A.3: *Audio Features Summarizer Sample#1*

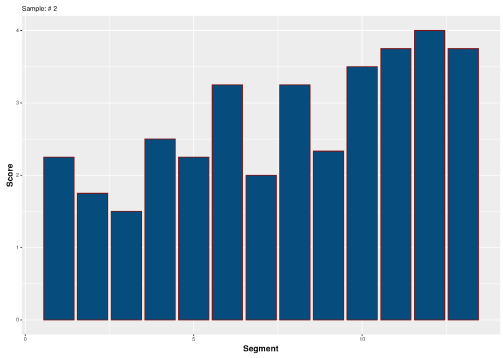

Figure A.4: *Audio Features Summarizer Sample#2*

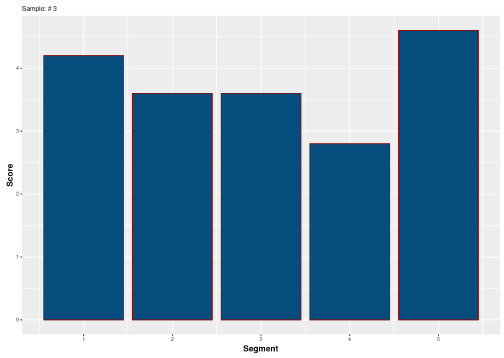

Figure A.5: *Audio Features Summarizer Sample#3*

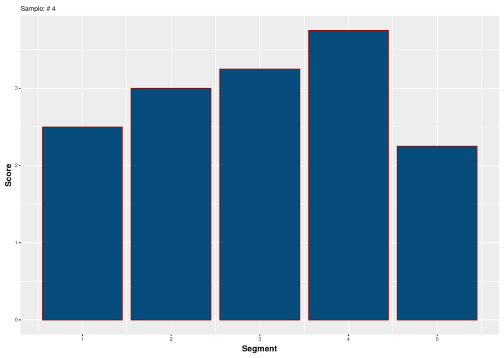

Figure A.6: *Audio Features Summarizer Sample#4*

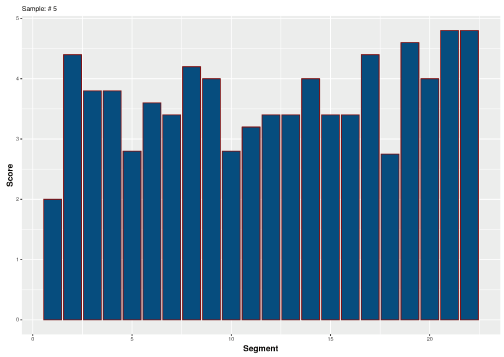

Figure A.7: *Audio Features Summarizer Sample#5*

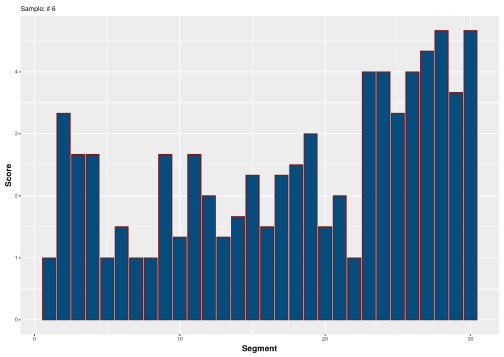

Figure A.8: *Audio Features Summarizer Sample#6*

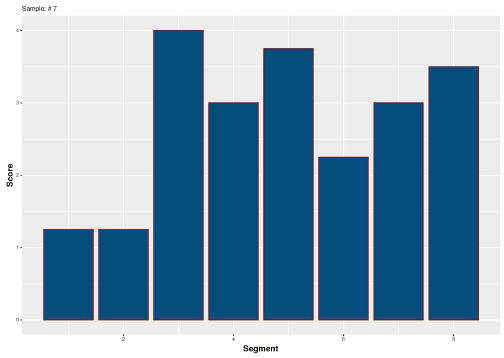

Figure A.9: *Audio Features Summarizer Sample#7*

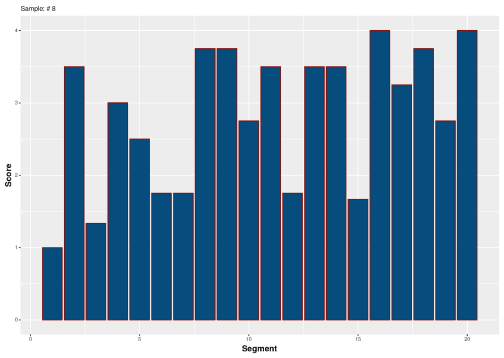

Figure A.10: *Audio Features Summarizer Sample#8*

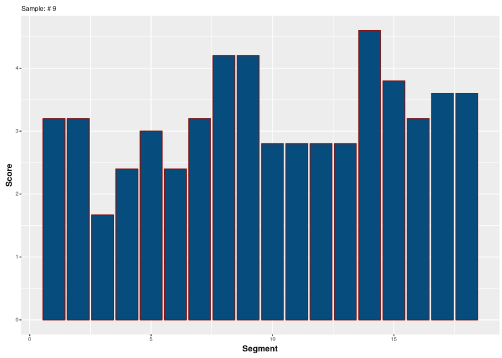

Figure A.11: *Audio Features Summarizer Sample#9*

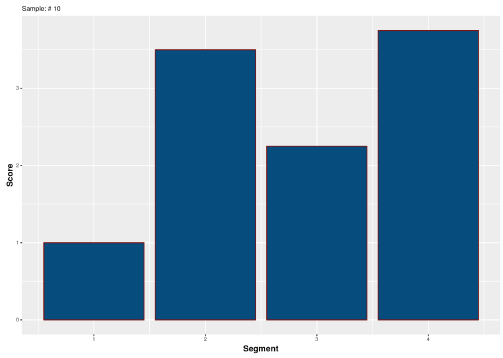

Figure A.12: *Audio Features Summarizer Sample#10*

Results regarding the Informativeness Model Summarizer described in Section 5.2 are shown from Figure A.13 to A.22. Blue rectangles correspond to each one of the segments of a sample. Score of each segment is represented by its height.

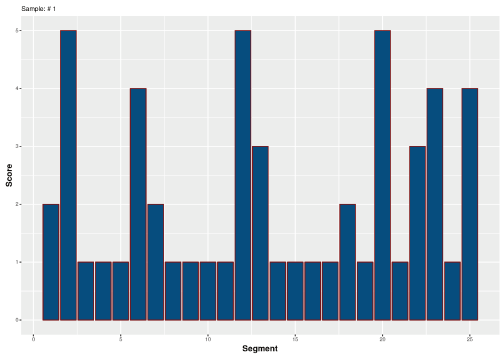

Figure A.13: *Informativeness Model Summarizer Sample#1*

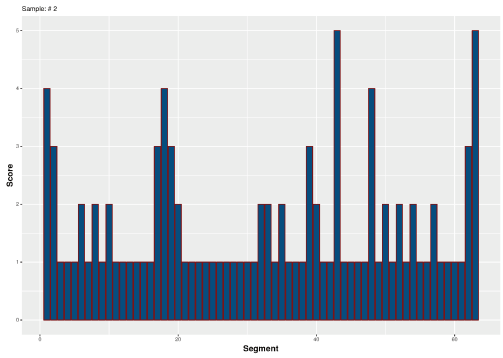

Figure A.14: *Informativeness Model Summarizer Sample#2*

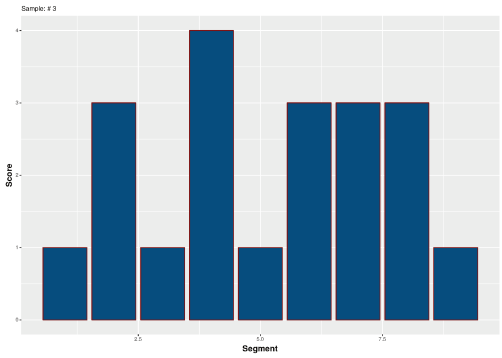

Figure A.15: *Informativeness Model Summarizer Sample#3*

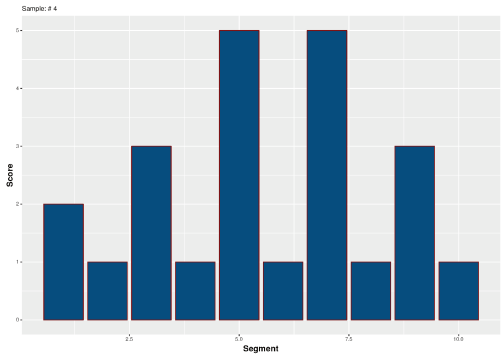

Figure A.16: *Informativeness Model Summarizer Sample#4*

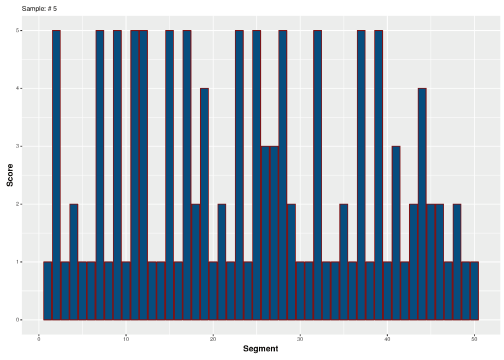

Figure A.17: *Informativeness Model Summarizer Sample#5*

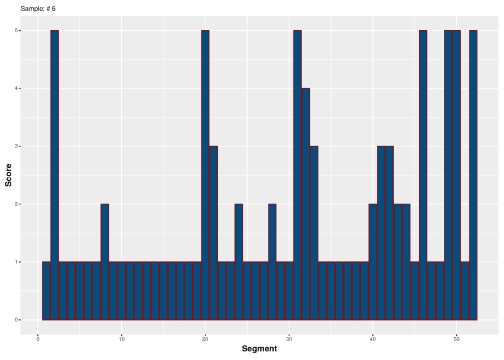

Figure A.18: *Informativeness Model Summarizer Sample #6*

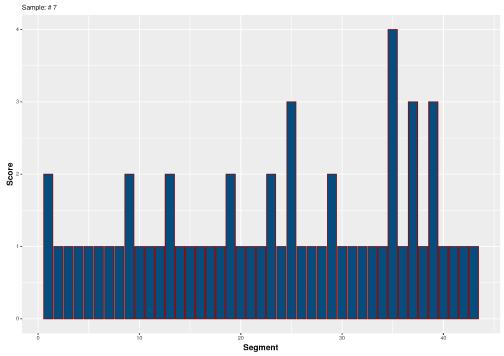

Figure A.19: *Informativeness Model Summarizer Sample#7*

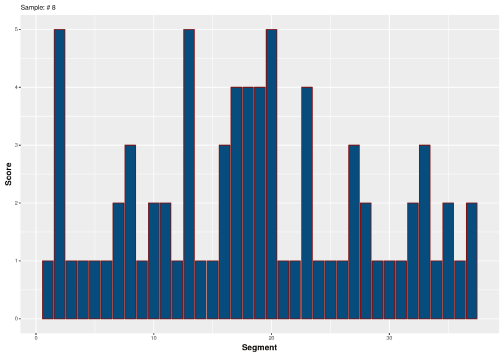

Figure A.20: *Informativeness Model Summarizer Sample#8*

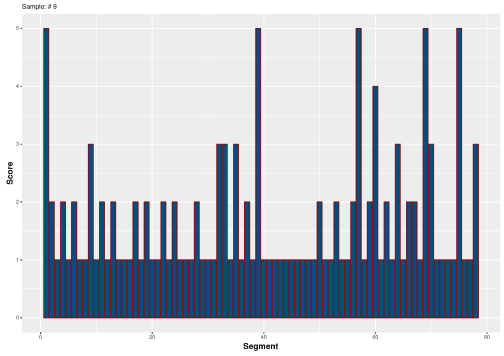

Figure A.21: *Informativeness Model Summarizer Sample#9*

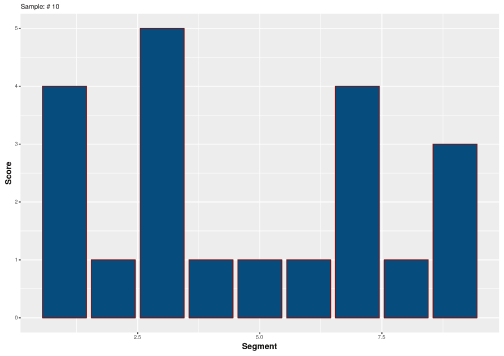

Figure A.22: *Informativeness Model Summarizer Sample#10*

Glossary

ALASR Arabic Loria Automatic Speech Recognition.

AMIS Access Multilingual Information opinionS.

ANN Artificial Neural Network.

ARTEX Autre Résumeur de TEXtes.

ARTEX-MSA ARTEX for Modern Standard Arabic.

ASR Automatic Speech Recognition.

ATS Automatic Text Summarization.

CER Classification Error Rate.

CNN Convolutional Neural Network.

CR Compression Ratio.

DM Data Mining.

FFNN Feedforward Neural Network.

FR Focused Retrieval.

FRESA FRamework for Evaluating Summaries Automatically.

HMM Hidden Markov Model.

IR Information Retrieval.

ITU Informative Text Unit.

JS Jensen-Shannon.

KL Kullback-Leibler.

LDC Linguistic Data Consortium.

LIA Laboratoire Informatique d'Avignon.

LM Language Model.

LogSim Logarithm Similarity.

LSA Latent Semantic Analysis.

LSTM Long Short Term Memory.

MER Match Error Rate.

MFCC Mel-frequency Cepstral Coefficient.

MSA Modern Standard Arabic.

MT Machine Translation.

nCG normalized Cumulative Gain.

NER Named-entity Recognition.

NLP Natural Language Processing.

NN Neural Network.

OOV out-of-vocabulary.

PCC Pearson Correlation Coefficient.

PMD Punctuation Marks Disambiguation.

POS Part-of-Speech.

QA Question Answering.

ReLU Rectified Linear Unit.

RIL Relative Information Loss.

RMS Root Mean Square.

RNN Recurrent Neural Network.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

SBD Sentence Boundary Detection.

SCU Summary Content Unit.

SER Slot Error Rate.

SU Semantic Unit.

TAC Text Analysis Conference.

VSM Vector Space Model.

WER Word Error Rate.

WIL Word Information Loss.

WiSeBE Window-based Sentence Boundary Evaluation.

List of Figures

List of Tables

Bibliography

Al-Abdallah, R. Z. & A. T. Al-Taani (2017). Arabic single-document text summarization using particle swarm optimization algorithm. *Procedia Computer Science 117*, 30–37.

Al-Khawaldeh, F. & V. Samawi (2015). Lexical cohesion and entailment based segmentation for arabic text summarization (lceas). *The World of Computer Science and Information Technology Journal (WSCIT) 5*(3), 51–60.

Al Qassem, L. M., D. Wang, Z. Al Mahmoud, H. Barada, A. Al-Rubaie, & N. I. Almoosa (2017). Automatic Arabic summarization: a survey of methodologies and systems. *Procedia Computer Science 117*, 10–18.

Al-Radaideh, Q. & M. Afif (2009). Arabic text summarization using aggregate similarity. In *International Arab conference on information technology (ACIT2009), Yemen*.

Al-Radaideh, Q. A. & D. Q. Bataineh (2018). A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms. *Cognitive Computation 10*(4), 651–669.

Al-Saleh, A. B. & M. E. B. Menai (2016). Automatic Arabic text summarization: a survey. *Artificial Intelligence Review 45*(2), 203–234.

Albelwi, S. & A. Mahmood (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy 19*(6), 242.

AlHanai, T., W.-N. Hsu, & J. Glass (2016). Development of the MIT ASR system for the 2016 Arabic multi-genre broadcast challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 299–304. IEEE.

Alotaiby, F., S. Foda, & I. Alkharashi (2010). Clitics in Arabic language: a statistical study. In *24th Pacific Asia Conference on Language, Information and Computation*.

Alsharhan, E. & A. Ramsay (2017). Improved Arabic speech recognition system through the automatic generation of fine-grained phonetic transcriptions. *Information Processing & Management*.

Althobaiti, M., U. Kruschwitz, & M. Poesio (2014). Aranlp: A java-based library for the processing of arabic text. In *LREC*.

Amini, M.-R. & E. Gaussier (2013). *Recherche d'information: Applications, modèles et algorithmes-Fouille de données, décisionnel et big data*. Editions Eyrolles.

Attia, M. & H. Somers (2008). *Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation*, Volume 279. University of Manchester Manchester.

Azmi, A. M. & S. Al-Thanyyan (2012). A text summarizer for Arabic. *Computer Speech & Language 26*(4), 260–273.

Azmi, A. M. & N. I. Altmami (2018). An abstractive Arabic text summarizer with user controlled granularity. *Information Processing & Management 54*(6), 903–921.

Baldridge, J. (2005). The OpenNLP project. http://opennlp.apache.org/.

Bellot, P., V. Moriceau, J. Mothe, E. SanJuan, & X. Tannier (2016). INEX Tweet Contextualization task: Evaluation, results and lesson learned. *Information Processing and Management 52*(5), 801–819.

Ben Jannet, M. A., M. Adda-Decker, O. Galibert, J. Kahn, & S. Rosset (2014). How to assess the quality of automatic transcriptions for the extraction of named entities? In *XXXe Journées d'Études sur la Parole (JEP'14)*, Le Mans, France, pp. 430–437.

Binwahlan, M. S. (2015). Extractive Summarization Method for Arabic Text-ESMAT. *International Journal of Computer Trends and Technology (IJCTT) 21*(2).

Blei, D. M., A. Y. Ng, & M. I. Jordan (2003). Latent dirichlet allocation. *Journal of machine Learning research 3*(Jan), 993–1022.

Bohac, M., K. Blavka, M. Kucharova, & S. Skodova (2012). Post-processing of the recognized speech for web presentation of large audio archive. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pp. 441–445. IEEE.

Bojanowski, P., E. Grave, A. Joulin, & T. Mikolov (2016). Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

Bridle, J. S. & M. D. Brown (1974). An experimental automatic word recognition system. *JSRU Report 1003*(5), 33.

Brum, H., F. Araujo, & F. Kepler (2016). Sentiment analysis for brazilian portuguese over a skewed class corpora. In *International Conference on Computational Processing of the Portuguese Language*, pp. 134–138. Springer.

Cabrera-Diego, L. A. & J.-M. Torres-Moreno (2018). SummTriver: A new trivergent model to evaluate summaries automatically without human references. *Data Knowledge Engineering 113*, 184 – 197.

Che, X., S. Luo, H. Yang, & C. Meinel (2016a). Sentence Boundary Detection Based on Parallel Lexical and Acoustic Models. In *Interspeech*, pp. 2528–2532.

Che, X., C. Wang, H. Yang, & C. Meinel (2016b). Punctuation Prediction for Unsegmented Transcript Based on Word Vector. In *LREC*.

Christensen, H., Y. Gotoh, & S. Renals (2008). A cascaded broadcast news highlighter. *IEEE transactions on audio, speech, and language processing 16*(1), 151–161.

Cité de l'Économie (2019). Invention de la télévision. `https://www.citeco.fr/10000-ans-histoire-economie/revolutions-industrielles/invention-de-la-television`. [Online; accessed 12-July-2019].

Dang, H. T. (2005). Overview of DUC 2005. In *Proceedings of the Document Understanding Conference*, pp. 1–12.

Dang, H. T., D. Kelly, & J. Lin (2007). Overview of the TREC 2007 Question Answering Track. In *Text Retrieval Conference (TREC)*.

Dang, H. T. & K. Owczarzak (2008). Overview of the tac 2008 opinion question answering and summarization tasks. In *Proc. of the First Text Analysis Conference*, Volume 2.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, & R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American society for information science 41*(6), 391–407.

Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, Volume 110.

Ding, D., F. Metze, S. Rawat, P. F. Schulam, S. Burger, E. Younessian, L. Bao, M. G. Christel, & A. Hauptmann (2012). Beyond Audio and Video Retrieval: Towards Multimedia Summarization. In *2Nd ACM*, ICMR '12, New York, NY, USA, pp. 2:1–2:8. ACM.

Duxans, H., X. Anguera, & D. Conejero (2009). Audio based soccer game summarization. In *Broadband Multimedia Systems and Broadcasting, 2009. BMSB'09. IEEE International Symposium on*, pp. 1–6. IEEE.

Ekstrand-Abueg, M., V. Pavlu, & J. A. Aslam (2013). Live nuggets extractor: a semi-automated system for text extraction and test collection creation. In *SIGIR*, pp. 1087–1088.

El-Haj, M., U. Kruschwitz, & C. Fox (2010). Using Mechanical Turk to create a corpus of Arabic summaries.

El-Haj, M., U. Kruschwitz, & C. Fox (2011). Exploring clustering for multi-document arabic summarisation. In *Asia Information Retrieval Symposium*, pp. 550–561. Springer.

El-Khair, I. A. (2006). Effects of stop words elimination for Arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences 4*(3), 119–133.

El-Masri, M., N. Altrabsheh, H. Mansour, & A. Ramsay (2017). A web-based tool for Arabic sentiment analysis. *Procedia Computer Science 117*, 38–45.

Farghaly, A. & K. Shaalan (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP) 8*(4), 14.

Fastl, H. & E. Zwicker (2006). *Psychoacoustics: facts and models*, Volume 22. Springer Science & Business Media.

Ferrucci, D. & A. Lally (2004). UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering 10*(3-4), 327–348.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin 76*(5), 378.

Fohr, D., O. Mella, & I. Illina (2017). New Paradigm in Speech Recognition: Deep Neural Networks. In *IEEE International Conference on Information Systems and Economic Intelligence*.

Giannakopoulos, G., M. El-Haj, B. Favre, M. Litvak, J. Steinberger, & V. Varma (2011). TAC2011 MultiLing Pilot Overview. In *4th Text Analysis Conference TAC*.

Goodfellow, I., Y. Bengio, & A. Courville (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gotoh, Y. & S. Renals (2000). Sentence boundary detection in broadcast speech transcripts. In *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*.

Graff, D. (2006). French Gigaword First Edition LDC2006T17. DVD. Philadelphia: Linguistic Data Consortium.

Habash, N., O. Rambow, & R. Roth (2009). MADA+ TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *2nd International Conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, Volume 41, pp. 62.

Habash, N. Y. (2010). Introduction to Arabic natural language processing. *Synthesis Lectures on Human Language Technologies 3*(1), 1–187.

Haboush, A., M. Al-Zoubi, A. Momani, & M. Tarazi (2012). Arabic text summarization model using clustering techniques. *World of Computer Science and Information Technology Journal (WCSIT) ISSN*, 2221–0741.

Hadrich, L. B., L. Baccour, & G. Mourad (2005). STAr: un Système de Segmentation de Textes Arabes basé sur l'analyse contextuelle des signes de ponctuations et de certaines particules. In *TALN'05*.

Harrag, F., A. Hamdi-Cherif, & A. Salman Al-Salman (2010). Comparative study of topic segmentation Algorithms based on lexical cohesion: Experimental results on Arabic language. *Arabian Journal for Science and Engineering 35*(2), 183.

Harris, Z. S. (1954). Distributional structure. *Word 10*(2-3), 146–162.

Hearst, M. A. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics 23*(1), 33–64.

Hebb, D. O. (1949). *The organization of behavior: a neuropsychological theory*. Science Editions.

Hertz, J. A. (2018). *Introduction to the theory of neural computation*. CRC Press.

Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine 29*(6), 82–97.

Holzinger, A., B. Malle, P. Kieseberg, P. M. Roth, H. Müller, R. Reihs, & K. Zatloukal (2017). Machine learning and knowledge extraction in digital pathology needs an integrative approach. In *Towards Integrative Machine Learning and Knowledge Extraction*, pp. 13–50. Springer.

Igras, M. & B. Ziółko (2016). Detection of sentence boundaries in Polish based on acoustic cues. *Archives of Acoustics 41*(2), 233–243.

Jaafar, Y. & K. Bouzoubaa (2018). A Survey and Comparative Study of Arabic NLP Architectures. In *Intelligent Natural Language Processing: Trends and Applications*, pp. 585–610. Springer.

Jouvet, D., D. Langlois, M. Menacer, D. Fohr, O. Mella, & K. Smaïli (2018). Adaptation of speech recognition vocabularies for improved transcription of YouTube videos. *Journal of the International Science and General Applications 1*(1), 1–9.

Kamps, J., S. Geva, & A. Trotman (2008). Report on the SIGIR 2008 workshop on focused retrieval. *ACM SIGIR Forum 42*(2), 59–65.

Kiss, T. & J. Strunk (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics 32*(4), 485–525.

Kolář, J. & L. Lamel (2012). Development and evaluation of automatic punctuation for French and English speech-to-text. In *Thirteenth Annual Conference of the International Speech Communication Association*.

LeCun, Y. et al. (1989). Generalization and network design strategies. In *Connectionism in perspective*, Volume 19. Citeseer.

Leszczuk, M., M. Grega, A. Koźbiał, J. Gliwski, K. Wasieczko, & K. Smaïli (2017). Video Summarization Framework for Newscasts and Reports – Work in Progress. In A. Dziech & A. Czyżewski (Eds.), *Multimedia Communications, Services and Security*, Cham, pp. 86–97. Springer International Publishing.

Lin, C.-Y. (1999). Training a Selection Function for Extraction. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, New York, NY, USA, pp. 55–62. ACM.

Lin, C.-Y. (2004). ROUGE: a package for automatic evaluation of summaries. pp. 74–81.

Lin, C.-Y., G. Cao, J. Gao, & J.-Y. Nie (2006). An information-theoretic approach to automatic evaluation of summaries. In *Conference on Human Language Technology Conference of the North American Chapter*, Morristown, NJ, Etats-Unis, pp. 463–470. ACL.

Lin, C.-Y. & E. H. Hovy (2003). Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *HLT-NAACL*, Volume 1, pp. 71–78.

Lin, C.-Y. & F. Och (2004). Looking for a few good metrics: ROUGE and its evaluation. In *NTCIR Workshop*.

Lin, J. J. & P. Zhang (2007). Deconstructing nuggets: the stability and reliability of complex question answering evaluation. In *SIGIR*, pp. 327–334.

Linhares Pontes, E., S. Huet, J.-M. Torres-Moreno, & A. C. Linhares (2018). Cross-Language Text Summarization Using Sentence and Multi-Sentence Compression. In M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, & F. Meziane (Eds.), *Natural Language Processing and Information Systems*, Cham, pp. 467–479. Springer International Publishing.

Liu, Y., N. V. Chawla, M. P. Harper, E. Shriberg, & A. Stolcke (2006). A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech & Language 20*(4), 468–494.

Logan, B. et al. (2000). Mel Frequency Cepstral Coefficients for Music Modeling. In *ISMIR*, Volume 270, pp. 1–11.

Louis, A. & A. Nenkova (2008). Automatic Summary Evaluation without Human Models. In *TAC*.

Louis, A. & A. Nenkova (2009). Automatically evaluating content selection in summarization without human models. In *2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 306–314. ACL.

Lu, W. & H. T. Ng (2010). Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pp. 177–186. Association for Computational Linguistics.

Mallek, F., N. T. Le, & F. Sadat (2018). Automatic Machine Translation for Arabic Tweets. In *Intelligent Natural Language Processing: Trends and Applications*, pp. 101–119. Springer.

Mani, I. & M. T. Maybury (2001). Automatic summarization.

Manning, C. D. & H. Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press.

Maskey, S. & J. Hirschberg (2005). Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Ninth European Conference on Speech Communication and Technology*.

Maskey, S. & J. Hirschberg (2006). Summarizing speech without text using hidden markov models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 89–92. Association for Computational Linguistics.
English

McCowan, I. A., D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner, & H. Bourlard (2004). On the Use of Information Retrieval Measures for Speech Recognition Evaluation. Idiap-RR Idiap-RR-73-2004, IDIAP, Martigny, Switzerland.

McCulloch, W. S. & W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5*(4), 115–133.

McFee, B., C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, & O. Nieto (2015). librosa: Audio and music signal analysis in python. In *14th python in science conference*, pp. 18–25.

Menacer, M. A., O. Mella, D. Fohr, D. Jouvet, D. Langlois, & K. Smaïli (2017a). Development of the Arabic Loria Automatic Speech Recognition system (ALASR) and its evaluation for Algerian dialect. *Procedia Computer Science 117*, 81–88.

Menacer, M. A., O. Mella, D. Fohr, D. Jouvet, D. Langlois, & K. Smaili (2017b). An enhanced automatic speech recognition system for Arabic. In *Third Arabic Natural Language Processing Workshop*, pp. 157–165.

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence 116*, 374–388.

Meteer, M. & R. Iyer (1996). Modeling conversational speech for speech recognition. In *Conference on Empirical Methods in Natural Language Processing*.

Mikolov, T., K. Chen, G. Corrado, & J. Dean (2013a). Efficient Estimation of Word Reresentations in Vector Space. *CoRR abs/1301.3781*.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, & J. Dean (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, T., W.-t. Yih, & G. Zweig (2013c). Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, Volume 13, pp. 746–751.

Miller, G. A. (1955). Note on the bias of information estimates. In *Information Theory in Psychology: Problems and Methods*, pp. 95–100.

Miniwatts Marketting Group (2019). Internet World Users by Language. `https://www.internetworldstats.com/stats7.htm`. [Online; accessed 21-September-2019].

Morris, A. C., V. Maier, & P. D. Green (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *INTERSPEECH*. ISCA.

Mrozinski, J., E. W. Whittaker, P. Chatain, & S. Furui (2006). Automatic sentence segmentation of speech for automatic summarization. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Volume 1, pp. I–I. IEEE.

Nenkova, A. & R. Passonneau (2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*, pp. 145–152.

Nenkova, A., R. Passonneau, & K. McKeown (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP) 4*(2), 4.

Ng, J.-P. & V. Abrecht (2015). Better summarization evaluation with word embeddings for rouge. *arXiv:1508.06034 [cs.CL]*.

Nicola, U., B. Maximilian, & V. Paul (2013). Improved models for automatic punctuation prediction for spoken and written text. In *Proceedings of INTERSPEECH*.

Oufaida, H., O. Nouali, & P. Blache (2014). Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization. *Journal of King Saud University-Computer and Information Sciences 26*(4), 450–461.

Palermo, Elizabeth (2014). Who Invented the Printing Press? `https://www.livescience.com/43639-who-invented-the-printing-press.html`. [Online; accessed 12-July-2019].

Palmer, D. D. & M. A. Hearst (1994). Adaptive Sentence Boundary Disambiguation. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, ANLC '94, Stroudsburg, PA, USA, pp. 78–83. Association for Computational Linguistics.

Palmer, D. D. & M. A. Hearst (1997). Adaptive Multilingual Sentence Boundary Disambiguation. *Comput. Linguist. 23*(2), 241–267.

Parker, R., D. Graff, K. Che, J. Kong, & K. Maeda (2011a). Arabic Gigaword Fifth Edition LDC2011T11. Web Download. Philadelphia: Linguistic Data Consortium.

Parker, R., D. Graff, J. Kong, K. Che, & K. Maeda (2011b). English Gigaword Fifth Edition LDC2011T07. Web Download. Philadelphia: Linguistic Data Consortium.

Pasha, A., M. Al-Badrashiny, M. T. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, & R. Roth (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *LREC*, Volume 14, pp. 1094–1101.

Pavlu, V., S. Rajput, P. B. Golbus, & J. A. Aslam (2012). IR system evaluation using nugget-based test collections. In *WSDM*, pp. 393–402.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London 58*, 240–242.

Peitz, S., M. Freitag, & H. Ney (2014). Better Punctuation Prediction with Hierarchical Phrase-Based Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT), South Lake Tahoe, CA, USA*.

Pennington, J., R. Socher, & C. D. Manning (2014). GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Povey, D., A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Number EPFL-CONF-192584. IEEE Signal Processing Society.

Qaroush, A., I. A. Farha, W. Ghanem, M. Washaha, & E. Maali (2019). An efficient single document Arabic text summarization using a combination of statistical and semantic features. *Journal of King Saud University-Computer and Information Sciences*.

Radev, D., A. Winkel, & M. Topper (2002). Multi document centroid-based text summarization. In *ACL 2002*. Citeseer.

Radev, D. R., S. Teufel, H. Saggion, W. Lam, J. Blitzer, H. Qi, A. Çelebi, D. Liu, & E. Drábek (2003). Evaluation Challenges in Large-Scale Document Summarization. In *ACL'03*, pp. 375–382.

Rafii, Z. & B. Pardo (2012). Music/Voice Separation Using the Similarity Matrix. In *ISMIR*, pp. 583–588.

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Rott, M. & P. Červa (2013). Summec: A summarization engine for czech. In *International Conference on Text, Speech and Dialogue*, pp. 527–535. Springer.

Rott, M. & P. Červa (2016). Speech-to-Text Summarization Using Automatic Phrase Extraction from Recognized Text. In P. Sojka, A. Horák, I. Kopeček, & K. Pala (Eds.), *Text, Speech, and Dialogue*, Cham, pp. 101–108. Springer International Publishing.

Rush, J. E., R. Salvador, & A. Zamora (1971). Automatic abstracting and indexing. II. Production of indicative abstracts by application of contextual inference and syntactic coherence criteria. *Journal of the American Society for Information Science 22*(4), 260–274.

Ryding, K. C. (2005). *A reference grammar of modern standard Arabic*. Cambridge university press.

Saggion, H. & G. Lapalme (2002). Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics 28*(4), 497–526.

Saggion, H. & T. Poibeau (2013). *Automatic Text Summarization: Past, Present and Future*, pp. 3–21. Berlin, Heidelberg: Springer Berlin Heidelberg.

Saggion, H., J.-M. Torres-Moreno, I. da Cunha, E. SanJuan, & P. Velázquez-Morales (2010). Multilingual Summarization Evaluation without Human Models. In *COLING*, pp. 1059–1067.

Sakai, T. & K. Sparck-Jones (2001). Generic Summaries for Indexing in Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, New York, NY, USA, pp. 190–198. ACM.

Salton, G., A. Wong, & C. S. Yang (1975). A Vector Space Model for Automatic Indexing. *Commun. ACM 18*(11), 613–620.

SanJuan, E., V. Moriceau, X. Tannier, P. Bellot, & J. Mothe (2012). Overview of the INEX 2012 Tweet Contextualization Track. In *CLEF (Working Notes/Labs/Workshop)*.

Saon, G., H.-K. J. Kuo, S. Rennie, & M. Picheny (2015). The IBM 2015 English conversational telephone speech recognition system. *arXiv preprint arXiv:1505.05899*.

Shriberg, E. & A. Stolcke (1996). Word predictability after hesitations: a corpus-based study. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, Volume 3, pp. 1868–1871. IEEE.

Silberztein, M. (2005). NooJ: a linguistic annotation system for corpus processing. In *HLT/EMNLP on Interactive Demonstrations*, pp. 10–11. Association for Computational Linguistics.

Smaïli, K., D. Fohr, C. González-Gallardo, M. Grega, L. Janowski, D. Jouvet, A. Komorowski, A. Kozbial, D. Langlois, M. Leszczuk, O. Mella, M. A. Menacer, A. Mendez, E. Linhares Pontes, E. Sanjuan, D. Swist, J.-M. Torres-Moreno, & B. Garcia-Zapirain (2018). A First Summarization System of a Video in a Target Language. In *MISSI 2018 - 11th edition of the International Conference on Multimedia and Network Information Systems*, Wroclaw, Poland, pp. 1–12.

Souteh, Y. & K. Bouzoubaa (2011). SAFAR platform and its morphological layer. In *Eleventh Conference on Language Engineering ESOLEC*, pp. 14–15.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Sterling, Christopher H. and Skretvedt, Randy (2019). Radio. `https://www.britannica.com/topic/radio`. [Online; accessed 12-July-2019].

Stevenson, M. & R. Gaizauskas (2000). Experiments on sentence boundary detection. In *Proceedings of the sixth conference on Applied natural language processing*, pp. 84–89. Association for Computational Linguistics.

Stolcke, A. & E. Shriberg (1996). Automatic linguistic segmentation of conversational speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, Volume 2, pp. 1005–1008. IEEE.

Strassel, S. (2003). Simple Metadata Annotation Specification V5. 0, Linguistic Data Consortium. *http://www.ldc.upenn.edu/projects/MDE/Guidelines/SimpleMDE_V5.0.pdf*.

Szaszák, G. & A. Beke (2012). Exploiting prosody for automatic syntactic phrase boundary detection in speech. *Journal of Language Modeling 1*(0), 143–172.

Szaszák, G., M. Á. Tündik, & A. Beke (2016). Summarization of Spontaneous Speech using Automatic Speech Recognition and a Speech Prosody based Tokenizer. In *KDIR*, pp. 221–227.

Taskiran, C. M., A. Amir, D. B. Ponceleon, & E. J. Delp (2001). Automated video summarization using speech transcripts. In *Storage and Retrieval for Media Databases 2002*, Volume 4676, pp. 371–383. International Society for Optics and Photonics.

Taskiran, C. M., Z. Pizlo, A. Amir, D. Ponceleon, & E. J. Delp (2006). Automated video program summarization using speech transcripts. *IEEE Transactions on Multimedia 8*(4), 775–791.

Teager, H. & S. Teager (1990). Evidence for nonlinear sound production mechanisms in the vocal tract. In *Speech production and speech modelling*, pp. 241–261. Springer.

Tilk, O. & T. Alumäe (2016). Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration. In *Interspeech 2016*.

Tomashenko, N., K. Vythelingum, A. Rousseau, & Y. Estève (2016). LIUM ASR systems for the 2016 Multi-Genre Broadcast Arabic challenge. In *Spoken Language Technology Workshop (SLT), 2016*, pp. 285–291. IEEE.

Torres-Moreno, J. (2015). Trivergence of Probability Distributions, at glance. *CoRR abs/1506.06205*.

Torres-Moreno, J., H. Saggion, I. da Cunha, E. SanJuan, & P. Velázquez-Morales (2010). Summary Evaluation with and without References. *Polibits 42*, 13–19.

Torres-Moreno, J.-M. (2012a). Artex is another text summarizer. *arXiv preprint arXiv:1210.3312*.

Torres-Moreno, J.-M. (2012b). Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. *arXiv preprint arXiv:1209.3126*.

Torres-Moreno, J.-M. (2014). *Automatic text summarization*. John Wiley & Sons.

Tran, N.-T., V.-T. Luong, N. L.-T. Nguyen, & M.-Q. Nghiem (2016). Effective Attention-based Neural Architectures for Sentence Compression with Bidirectional Long Short-term Memory. In *Seventh Symposium on Information and Communication Technology*, SoICT '16, New York, NY, USA, pp. 123–130. ACM.

Treviso, M. V., C. D. Shulby, & S. M. Aluisio (2017). Evaluating Word Embeddings for Sentence Boundary Detection in Speech Transcripts. *arXiv preprint arXiv:1708.04704*.

Van Rijsbergen, C. J., S. E. Robertson, & M. F. Porter (1980). *New models in probabilistic information retrieval*. British Library Research and Development Department London.

Vassilakis, P. N. (2001). *Perceptual and physical properties of amplitude fluctuation and their musical significance*. Ph. D. thesis, University of California, Los Angeles.

Vu-Quoc, Loc (2016). Neuron3. `https://commons.wikimedia.org/wiki/File:Neuron3.png`. [Online; accessed 19-October-2019].

Wang, W., G. Tur, J. Zheng, & N. F. Ayan (2010). Automatic disfluency removal for improving spoken language translation. In *Acoustics speech and signal processing (icassp), 2010 ieee international conference on*, pp. 5214–5217. IEEE.

Yilmaz, E., M. Verma, R. Mehrotra, E. Kanoulas, B. Carterette, & N. Craswell (2015). Overview of the TREC 2015 Tasks Track. In *Text Retrieval Conference (TREC)*.

Yu, D. & L. Deng (2016). *Automatic speech recognition.* Springer.

Zechner, K. (2003). Spoken language condensation in the 21st century. In *Eighth European Conference on Speech Communication and Technology*.

Zlatintsi, A., E. Iosif, P. Marago, & A. Potamianos (2015). Audio salient event detection and summarization using audio and text modalities. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pp. 2311–2315. IEEE.

Zlatintsi, A., P. Maragos, A. Potamianos, & G. Evangelopoulos (2012). A saliency-based approach to audio event detection and summarization. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1294–1298. IEEE.

Zribi, I., I. Kammoun, M. Ellouze, L. Belguith, & P. Blache (2016). Sentence boundary detection for transcribed Tunisian Arabic. *Bochumer Linguistische Arbeitsberichte*, 323.

Personal Bibliography

González-Gallardo, C.-E., R. Deveaud, E. Sanjuan, & J.-M. Torres-Moreno (2019). Audio Summarization with Audio Features and Probability Distribution Divergence. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*, La Rochelle, France.

González-Gallardo, C.-E., M. Hajjem, E. Sanjuan, & J.-M. Torres-Moreno (2018a). Étude de l'informativité des transcriptions : une approche basée sur le résumé automatique. In *Conférence en Recherche d'Information et Applications (CORIA)*, Rennes, France.

González-Gallardo, C.-E., E. L. Pontes, F. Sadat, & J.-M. Torres-Moreno (2018b). Automated Sentence Boundary Detection in Modern Standard Arabic Transcripts using Deep Neural Networks. *Procedia Computer Science 142*, 339–346.

González-Gallardo, C.-E., E. SanJuan, & J.-M. Torres-Moreno (2018c). Extending Text Informativeness Measures to Passage Interestingness Evaluation Language Model vs. Word Embedding. *Int. J. Comput. Linguistics Appl. 10*(1), XX–XX.

González-Gallardo, C.-E. & J.-M. Torres-Moreno (2018a). Sentence Boundary Detection for French with Subword-Level Information Vectors and Convolutional Neural Networks. *arXiv preprint arXiv:1802.04559*.

González-Gallardo, C.-E. & J.-M. Torres-Moreno (2018b). WiSeBE: Window-Based Sentence Boundary Evaluation. In *Mexican International Conference on Artificial Intelligence*, pp. 119–131. Springer.

Grega, M., K. Smaïli, M. Leszczuk, C.-E. González-Gallardo, J.-M. Torres-Moreno, E. Linhares Pontes, D. Fohr, O. Mella, M. Menacer, & D. Jouvet (2019). An Integrated AMIS Prototype for Automated Summarization and Translation of Newscasts and Reports. In K. Choroś, M. Kopel, E. Kukla, & A. Siemiński (Eds.), *Multimedia and Network Information Systems*, Cham, pp. 415–423. Springer International Publishing.

Menacer, M. A., C.-E. González-Gallardo, K. Abidi, D. Fohr, D. Jouvet, D. Langlois, O. Mella, F. Sadat, J.-M. Torres-Moreno, & K. Smaïli (2019). Extractive Text-Based Summarization of Arabic Videos: Issues, Approaches and Evaluations. In *International Conference on Arabic Language Processing*, pp. 65–78. Springer.

Pontes, E. L., C.-E. González-Gallardo, J.-M. Torres-Moreno, & S. Huet (2018). Cross-lingual speech-to-text summarization. In *International Conference on Multimedia and Network Information System*, pp. 385–395. Springer.

Smaïli, K., D. Fohr, C. González-Gallardo, M. Grega, L. Janowski, D. Jouvet, A. Komorowski, A. Kozbial, D. Langlois, M. Leszczuk, O. Mella, M. A. Menacer, A. Mendez, E. Linhares Pontes, E. Sanjuan, D. Swist, J.-M. Torres-Moreno, & B. Garcia-Zapirain (2018). A First Summarization System of a Video in a Target Language. In *MISSI 2018 - 11th edition of the International Conference on Multimedia and Network Information Systems*, Wroclaw, Poland, pp. 1–12.

Smaïli, K., D. Fohr, C.-E. González-Gallardo, M. Grega, L. Janowski, D. Jouvet, A. Koźbiał, D. Langlois, M. Leszczuk, O. Mella, et al. (2019). Summarizing videos into a target language: Methodology, architectures and evaluation. *Journal of Intelligent & Fuzzy Systems* (Preprint), 1–12.