Compressive Cross-Language Text Summarization

Elvys Linhares Pontes

ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THESIS

presented at the Université d'Avignon et des Pays de Vaucluse
to obtain the degree of Doctor

SPECIALITY : Computer Science

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

Compressive Cross-Language Text Summarization

by
Elvys LINHARES PONTES

Defended publicly on November 30, 2018 before the jury members:

M^{me}	Marie-Francine MOENS	Professor, LIIR, Heverlee	Rapporteur
M.	Antoine DOUCET	Professor, L3i, La Rochelle	Rapporteur
M.	Frédéric BECHET	Professor, LIS, Marseille	Examiner
M.	Guy LAPALME	Professor, DIRO, Montréal	Examiner
M^{me}	Fatiha SADAT	Professor, GDAC, Montréal	Examiner
M.	Petko VALTCHEV	Professor, GDAC, Montréal	Examiner
M.	Florian BOUDIN	Associate Professor, LS2N, Nantes	Examiner
M.	Juan-Manuel TORRES-MORENO	Associate Professor HDR, LIA, Avignon	Advisor
M.	Stéphane HUET	Associate Professor, LIA, Avignon	Co-Advisor
M^{me}	Andréa Carneiro LINHARES	Associate Professor, UFC, Fortaleza	Co-Advisor

---

Laboratoire Informatique d'Avignon

Abstract

The popularization of social networks and digital documents increased quickly the information available on the Internet. However, this huge amount of data cannot be analyzed manually. Natural Language Processing (NLP) analyzes the interactions between computers and human languages in order to process and to analyze natural language data. NLP techniques incorporate a variety of methods, including linguistics, semantics and statistics to extract entities, relationships and understand a document. Among several NLP applications, we are interested, in this thesis, in the cross-language text summarization which produces a summary in a language different from the language of the source documents. We also analyzed other NLP tasks (word encoding representation, semantic similarity, sentence and multi-sentence compression) to generate more stable and informative cross-lingual summaries.

Most of NLP applications (including all types of text summarization) use a kind of similarity measure to analyze and to compare the meaning of words, chunks, sentences and texts in their approaches. A way to analyze this similarity is to generate a representation for these sentences that contains the meaning of them. The meaning of sentences is defined by several elements, such as the context of words and expressions, the order of words and the previous information. Simple metrics, such as cosine metric and Euclidean distance, provide a measure of similarity between two sentences; however, they do not analyze the order of words or multi-words. Analyzing these problems, we propose a neural network model that combines recurrent and convolutional neural networks to estimate the semantic similarity of a pair of sentences (or texts) based on the local and general contexts of words. Our model predicted better similarity scores than baselines by analyzing better the local and the general meanings of words and multi-word expressions.

In order to remove redundancies and non-relevant information of similar sentences, we propose a multi-sentence compression method that compresses similar sentences by fusing them in correct and short compressions that contain the main information of these similar sentences. We model clusters of similar sentences as word graphs. Then, we apply an integer linear programming model that guides the compression of these clusters based on a list of keywords. We look for a path in the word graph that has good cohesion and contains the maximum of keywords. Our approach outperformed baselines by generating more informative and correct compressions for French, Portuguese and Spanish languages.

Finally, we combine these previous methods to build a cross-language text summarization system. Our system is an {English, French, Portuguese, Spanish}-to-{English, French} cross-language text summarization framework that analyzes the information in both languages to identify the most relevant sentences. Inspired by the compressive text summarization methods in monolingual analysis, we adapt our multi-sentence compression method for this problem to just keep the main information. Our system proves to be a good alternative to compress redundant information and to preserve relevant information. Our system improves informativeness scores without losing grammatical quality for French-to-English cross-lingual summaries. Analyzing {English, French, Portuguese, Spanish}-to-{English, French} cross-lingual summaries, our system significantly outperforms extractive baselines in the state of the art for all these languages. In addition, we analyze the cross-language text summarization of transcript documents. Our approach achieved better and more stable scores even for these documents that have grammatical errors and missing information.

Contents

Chapter 1

Introduction

Contents

Technological advance has improved and increased the speed of world communication through the transmission of videos, images and audios. Nowadays, most books and newspapers have digital and/or audio versions while the popularization of social networks (such as Facebook, Twitter, YouTube, among others) and news Web sites have enabled a great increase in the amount of data trafficked over the Internet about the most diverse subjects. Every day, a considerable amount of information is published in various sites, e.g. comments, photos, videos and audio in different languages. In this way, an event is quickly disseminated on the Web by different news sources from around the world and under various formats (audio, image, text and video).

Readers, besides not having the time to go through this amount of information, are not interested in all the proposed subjects and generally select the content of their interest. Another limiting factor is the language of messages, a lot of news being available in languages that readers do not know or have little knowledge of. It is worth mentioning that much of the information is personal, such as comments of daily life, personal photos and videos posted on social networks and blogs. Thus, some of this information is not of interest to everybody. For this reason, newspapers, movies, books, magazines, websites and blogs have headlines, summaries and/or synopses of the topics covered (Figure 1.1). Readers, from the headlines of a newspaper, identify the subject of news and then can choose which article to read in its entirety. This process is similar for books and movies with their synopses and descriptions on websites and blogs. In this way, readers can quickly identify the subject of their interest and then continue the reading. These synopses, descriptions and headlines are different types of summaries that highlight the main information of books and articles at different levels of granularities.

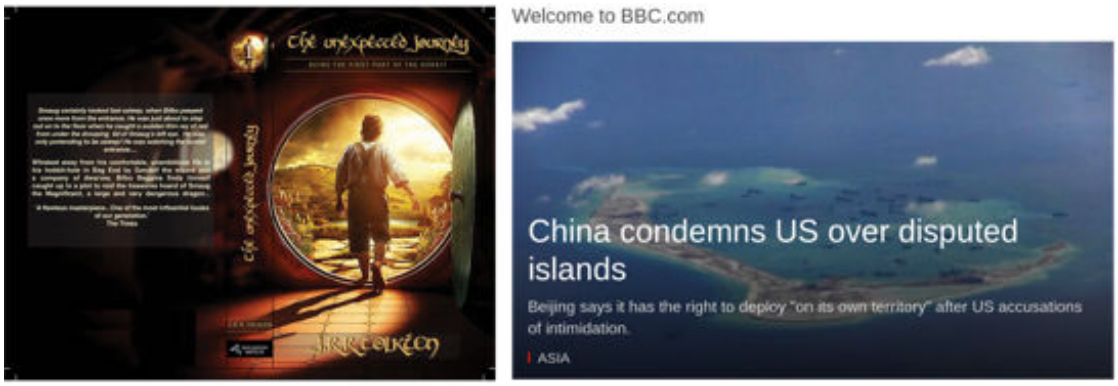


Figure 1.1: *Examples of summaries: book and website.*

In general, a summary is composed of the main idea presented in the original document in a short and objective way. For a better understanding of the word “summary", we present some definitions found in the literature and in dictionaries:

- A brief statement or account of the main points of something[1].
- A comprehensive and usually brief abstract, recapitulation, or compendium of previously stated facts or statements[2].
- The essential contents of a particular knowledge record and a real substitute for the document (Cleveland and Cleveland, 1983).
- A condensed version of a source document having a recognizable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source (Saggion and Lapalme, 2002).

A summary can be generated using several strategies; each one creates a different kind of summary with specific characteristics (Saggion and Lapalme, 2002; Moens et al., 2004). People normally use the following methodology to create a summary. Initially, they read a text and select the key information. With these data and their knowledge, they construct new shorter sentences containing the relevant information and the general idea contained in original texts. Another key feature of a summary is its length. The summary can be produced in different lengths depending on the desired end. For example, news headlines from newspapers and sites have few words to catch the reader’s attention and convey the key idea of the news. However, longer texts require a more comprehensive summary for the reader to understand the subject of the text as is the case of books, which requires longer summaries than daily news to get their general idea conveyed. One way to measure the length of a summary is its number of words or characters. Another possible way is the Compression Ratio (CR), which is responsible for defining the size of the summary in relation to its original text. CR is defined by the length of the summary over the length of the document (Equation 1.1).

[1]Source: `https://en.oxforddictionaries.com`
[2]Source: `https://www.dictionary.com`

$$\mathrm{CR} = \frac{|\text{summary}|}{|\text{document}|} \tag{1.1}$$

Summary can be considered as a kind of compression process that removes non-relevant content and maintains key text information. The lower the CR value, the shorter the summary of an analyzed text. This reduction, up to a certain level, improves the quality of a summary because it highlights the main information. However, the exaggerated reduction of a document causes the loss of relevant information and damages its comprehensibility.

Besides the length, summaries must contain relevant information. However, the relevance of sentences depends on the subject of documents, the context and the type of the summary. Summarizer systems attempt to identify the most relevant sentences in documents from the most discussed subjects. Nevertheless, some documents are composed of several subjects making the system consider sentences about different subjects with similar importance. In this case, summaries are composed of several subjects and may have less information about the "real" relevant information. The use of queries can be an alternative to mitigate this problem. Queries provide a context to classify the information as relevant or not for a specific summary. Some systems use the title or the first sentence of documents as queries to guide the summarization of these documents (Torres-Moreno, 2014).

Summaries facilitate and accelerate the acquisition of relevant information to the reader. However, the large number of texts and the high cost of professional summarizers make it impossible to summarize many documents within a reasonable time and affordable cost without the help of processing tools.

1.1 Cross-Language Text Summarization

Cross-Language Text Summarization (CLTS) aims to generate a summary of a document where the summary language differs from the document language. More precisely, CLTS consists in analyzing a document in a language source to get its meaning and, then, generate a short, informative and correct summary of this document in a target language. This process can be split in two main processes: text summarization and text translation. Two simple possible procedures are: summarize the document and, then, translate the summary; or translate the document to the target language and, then, summarize the translated document (more details in Chapter 2).

As we have discussed before, the enormous amount of information prevents it from being summarized and translated by humans. Besides the problem of summarizing all these documents, the translation of documents into several languages requires polyglot translators. This process requires a lot of time and resources when there are a huge amount of data to be analyzed.

A solution to this problem is presented through the automatic analysis of the test

data and the automatic generation of cross-lingual summaries. The next section describes how we can automatically analyze text documents using language processing.

Natural Language Processing

Natural Language Processing (NLP) is a research area involving Linguistics, Artificial Intelligence and Computer Science, and studying the interactions between natural human language and machines. More specifically, NLP deals with understanding, analysis, manipulation, and/or generation of natural language by computers. Natural language refers to speech analysis in both audible speech, as well as text of a language. NLP systems capture meaning from an input of words and symbols in the form of a structured output, e.g. a tweet, a review, a document and so on. There are a variety of approaches for processing human language:

- Symbolic: this approach is based on rules and lexicons. More precisely, it analyzes a human language following rules of a language defined by linguistic experts.
- Statistical: this approach generates models based on statistics to recognize recurring patterns in large text corpora. Particularly, a model develops its own linguistic rules based on the identification of trends in large samples of texts.
- Connectionism: this approach uses mathematical models, known as connectionist networks or artificial neural networks, to study the human cognition (Hanson and Burr, 1990). These connectionist networks discover by themselves the rules of languages.

The process of language analysis can be divided into a number of levels: morphological, syntactic, semantic and pragmatic analysis (Indurkhya and Damerau (2010), Figure 1.2):

- Morphological analysis: examines how the parts of words (morphemes) combine to make words.
- Syntactic analysis: focuses on text at the sentence level. The meaning of a sentence is dependent on word order and its syntactic dependency.
- Semantic analysis: focuses on how the context of words within a sentence helps determine the meaning of words.
- Pragmatic analysis: defines the meaning of a text based on the context of words and sentences.

This thesis is composed of several methods that use different levels of analysis. More precisely, Chapters 3, 6 and Appendix A use the semantic analysis to verify the context of words and to predict the semantic similarity between pairs of sentences. Chapters 4, 5 and 6 employ the syntactic analysis to verify the structure of sentences and to compress them for the summarization of documents.

NLP techniques incorporate a variety of methods, including linguistics, semantics, statistics and machine learning to extract entities, relationships and to take into ac-

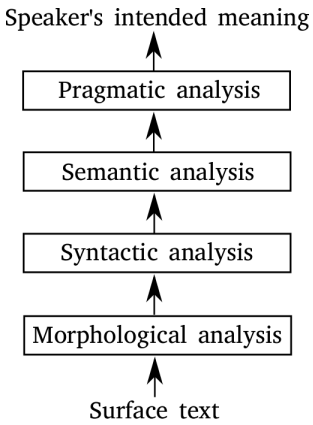


Figure 1.2: *The stages of a language analysis in processing natural language.*

count context. Rather than understanding isolated words or combinations of words, NLP helps computers understand the meaning of sentences and documents. It uses a number of methodologies to decipher ambiguities in language, including automatic summarization, part-of-speech tagging, disambiguation, entity extraction and relation extraction, as well as disambiguation and natural language understanding and recognition. Using this linguistic analysis, we can examine a document and calculate the similarity and relevance of its sentences. This document can also be automatically summarized and translated.

In order to generate more informative cross-lingual summaries, we investigated several NLP applications individually to build a modular CLTS framework. We investigated and developed new approaches for these applications using Neural Network (NN), Integer Linear Programming (ILP) and heuristic methods. Next section highlights our contributions for each NLP application and how we combined them to build our CLTS framework.

1.2 Objectives

The main objective of this thesis is to develop a framework to generate cross-lingual summaries of documents in {English, French, Portuguese, Spanish}-to-{English, French} languages. In order to accomplish this objective, we carried out an analysis of some NLP applications (sentence similarity, sentence compression and multi-sentence compression) to build our CLTS framework. In a formal way, the objectives of this thesis are:

- To provide a framework to analyze and to predict the semantic similarity of pairs of sentences in order to improve the analysis of documents and the clustering of similar sentences.
- To provide a sentence compression method to remove non-relevant information of sentences.

- To provide a method to compress similar sentences in order to reduce the redundancy of information in the documents.
- To build a modular framework using the previous methods to generate compressive cross-lingual summaries for {English, French, Portuguese, Spanish}-to-{English, French} languages.
- To carry out automatic and manual evaluations in several languages in order to analyze and compare the quality and the adaptability of our framework in relation to the state of the art.

1.3 Contributions

This thesis provides several contributions in sentence similarity, multi-sentence compression and CLTS. More precisely, Semantic Text Similarity (STS) aims to predict the degree of similarity between two sentences. Most works analyze only the general representation of words which does not represent the real meaning of these words in a sentence. A word has a specific meaning based on its previous and its following words in a sentence. Therefore, we propose a new NN model that combines Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) architectures to analyze local and general contexts of words in a sentence. This combination of contexts analyzes better the meaning of words and Multi-Word Expressions (MWEs) by providing a better semantic sentence representation and improving the prediction of sentence similarity.

Multi-Sentence Compression (MSC) combines the information of a cluster of similar sentences to generate a new informative and correct sentence. Several works (Filippova, 2010; Boudin and Morin, 2013) used the Word Graph model to represent a cluster of similar sentences and to generate a compression with the main information of these sentences. However, these works only use a simple cohesion measure (combination of the position, frequency and co-occurrence of words) to generate a compression. We devised a new Integer Linear Programming (ILP) method that uses keywords to guide the generation of compressions from a cluster of similar sentences. Our approach models a cluster of similar sentences as a Word Graph and calculates a path which is composed of words with a good cohesion and contains the biggest number of keywords of the cluster. We carry out an experiment on corpora in the French, Portuguese and Spanish languages to demonstrate the stability of our system with different languages and datasets.

Most works in CLTS used extractive approaches for summarization. Recently, some studies (Yao et al., 2015b; Zhang et al., 2016; Wan et al., 2018) have developed compressive and/or abstractive approaches; however, they are specific for a pair of languages, which limits the generation of cross-lingual summaries for other languages. We built on our work on MSC to propose a new compressive CLTS framework. We consider the information in the source and in the target languages to extract all relevant information available in both languages. Then, we proposed a combination of compres-

sive and extractive approaches to generate more informative French-to-English cross-lingual summaries. Our compressive approach combines a NN approach of the state of the art and our MSC model to compress sentences and similar sentences in a document, respectively. In order to validate the adaptability of our framework for other languages, we also carried out a multilingual analysis to evaluate {English, French, Portuguese, Spanish}-to-{English, French} cross-lingual summaries. Our compressive approach achieved better and more stable ROUGE scores than extractive baselines for all pairs of languages.

Finally, we propose two approaches for two CLTS applications: microblog contextualization and CLTS of transcripts. We combine Information Retrieval (IR), MSC and Text Summarization (TS) to retrieve relevant Wikipedia pages, to compress redundant information and to generate a summary that describes festivals in microblogs, respectively. Finally, we extend our compressive CLTS approach to analyze transcript documents. This type of documents contains grammatical errors and no punctuation marks. Our approach segments transcript documents and generates compressive cross-lingual summaries. Our method once again achieved better ROUGE scores than extractive baselines.

1.4 Structure of the Thesis

This thesis is organized as follows:

- Chapter 2 first makes an overview of neural network concepts and word representation to facilitate the comprehension of models described in the state of the art and our work. Then, we describe recent works of text summarization and machine translation to provide a background for the CLTS. Finally, we detail the most relevant works in the CLTS field by describing their approaches.

- Chapter 3 analyzes the semantic textual similarity that consists in determining a similarity score between two sentences. Our NN model combines general and local contexts. It uses a convolutional neural network to analyze the local context of words based on their previous and following words in a sentence. Then, a recurrent neural network analyzes both local and general contexts of words to improve the sentence analysis and to generate a semantic sentence representation. Finally, our model uses these representations to predict the semantic similarities between pairs of sentences.

- Chapter 4 presents the multi-sentence compression that consists in generating a single sentence with the main information of a cluster of similar sentences. We describe our approach that first models these clusters as word graphs. Then, an ILP model guides the compression of these clusters based on their keywords and their "cohesion" of words. We compared our model to other baselines on French, Portuguese and Spanish datasets. Our approach outperformed all baselines by generating more informative and correct compressions for all these languages.

- Chapter 5 deals with CLTS that produces a summary in a language different from the language of the source documents. We describe our new compressive CLTS framework that analyzes the text in both languages to calculate the relevance of sentences. Our approach compresses sentences at two levels: clusters of similar sentences are compressed using our MSC method and other sentences are compressed by a NN model. We carry out an analysis of {English, French, Portuguese, Spanish}-to-{English, French} cross-lingual summaries in order to compare the stability of our system with other extractive systems.
- Chapter 6 describes two CLTS applications. The first application is about the contextualization of microblogs using a Wikipedia dataset and the second one is about the CLTS of transcripts. In the latter application, our compressive CLTS approach achieved stable results and outperformed extractive CLTS approaches for transcript documents that are composed of several transcription and grammatical errors, and missing information.
- Chapter 7 contains the final conclusions about our work by describing advantages and limitations of our approaches. We also highlight the challenges of CLTS and we propose some future works about the evaluation and the generation of cross-lingual summaries.
- Appendix A describes a complementary work to this thesis on monolingual text summarization. We carry out an analysis of the relevance of word representation in the performance of monolingual text summarizer systems. We detail our approach using word embedding to create a discrete context vocabulary. Similar words being associated with a same representation vector. Extractive TS systems using our method outperformed the versions using one-hot encoding and word embedding for an English dataset and obtained better results than systems using one-hot encoding for a French dataset.

Chapter 2

State of the Art

Contents

Text representation is an essential step in NLP. The quality of document analysis depends on the amount of information that can be extracted from it. Among several types of text representations, word representations are widely used in the literature and can be split in two groups: one-hot encoding and word embeddings. On the one hand, one-hot encoding generates a simple representation for the words without considering their context or meaning. On the other hand, word embedding preserves the context of words, e.g. relationships of genre, syntactic, semantic, and so on. This context helps the sentence analysis and improves the representation of sentences and documents (Section 2.1). Recently, Neural Networks have played a leading role in NLP applications (including TS, Machine Translation (MT) and CLTS). They improved the analysis of

words, sentences and documents. The back-propagation algorithm among other improvements of learning techniques has enabled the exploitation of large datasets by Neural Networks models to improve the performance of several NLP systems. Therefore, we make an introduction about Neural Networks by describing their architectures and characteristics (Section 2.2).

CLTS is a complex NLP application involving TS and MT applications. Therefore, we make an overview of the most relevant approaches in TS, MSC and MT using different word encodings and Neural Networks architectures (Sections 2.3, 2.4 and 2.5) to provide a background for CLTS. Then, Section 2.6 details the last works in CLTS by highlighting their advantages and limitations. Finally, we position our work with respect to the state of the art in Section 2.7.

2.1 Text Representation

Most NLP applications depend on the representation of documents to analyze their content. There are several kinds of representations in the state of the art (Mikolov et al., 2013; Torres-Moreno, 2014; Chen et al., 2015). The bag-of-words is one of the most basic text representation where a document is represented by a matrix $\mathbf{D}^{[|S|\times|Voc|]}$. S is the set of sentences and Voc is the vocabulary of the document. The cell D_{ij} can be represented by: a binary to determine if a word j exists in the sentence i; the frequency of the word j in the sentence i or the distribution of the word j in the sentences (Term-Frequency Inverse Document Frequency, TF-IDF) (Spärck-Jones, 1972) Therefore, we can represent the document D by Equation 2.1 (Salton, 1968; Spärck-Jones, 1972; Torres-Moreno, 2014).

$$\mathbf{D} = \begin{pmatrix} D_{11} & D_{12} & \dots & D_{1|Voc|} \\ D_{21} & D_{22} & \dots & D_{2|Voc|} \\ \vdots & \vdots & & \vdots \\ D_{|S|1} & D_{|S|2} & \dots & D_{|S||Voc|} \end{pmatrix} \tag{2.1}$$

The bag-of-words can be used to represent which words exist in a sentence, paragraph or document. However, it does not retain the order of words in the sentences. Several works use different levels of representation (character, word, sentence, paragraph and document) to preserve the maximum of information (Mikolov et al., 2013; Le and Mikolov, 2014; Chen et al., 2015; Bojanowski et al., 2017a). Among them, some word representations consider the word as the smallest unit of the document. Other works combine one or more representations to generate more complex ones. For example, words can be represented by the combination of their characters (Chen et al., 2015), their morphemes (Botha and Blunsom, 2014), or their subwords (Bojanowski et al., 2017a). Le and Mikolov (2014) proposed an algorithm to represent sentences, paragraphs and documents by continuous vectors in order to predict words in the document.

The word representation is the most commonly used in the literature. Therefore, the following subsection details this representation by describing its advantages and disadvantages.

Word Representation

Word representation is a key feature for NLP to analyze and understand the content of a document. The token normalization can improve the word representation by reducing the complexity of text (Salton, 1968). This process consists in transforming all words into lower-case letters and/or substituting derivations according to their stem[1]. Other possibilities are filtering the document (e.g. remove stopwords and punctuation marks) and/or put all adjectives/nouns/verbs in a specific grammatical gender, number and tense (Torres-Moreno, 2014). The literature describes two kinds of word representations: one-hot encoding and continuous representation (or word embeddings). The following subsections describe these representations by highlighting their advantages and disadvantages.

One-hot Encoding

In order to analyze a document, we have to represent words in a mathematical way. One-hot encoding is the simplest way to represent these words. A one-hot encoding is a representation of categorical variables as binary vectors. Each integer word value is represented as a binary vector, that is all values are zero, except the index of the integer which is marked with 1 (Figure 2.1). For instance, a document with a vocabulary V uses a vector of N-dimension to represent each word of this vocabulary.

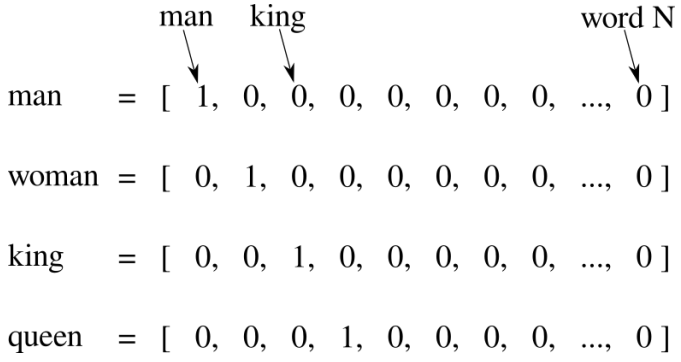


Figure 2.1: *One-hot encoding example.*

One-hot encoding is easy to implement; however, this representation does not preserve the context of words (all words are independent and they are at the same distance whatever their syntactic or semantic link). Moreover, this representation cannot be reused with other documents that have a different vocabulary.

[1]The stem of a word corresponds to the part of this word that never changes when inflected, e.g. "comput" is the stem of "computers", "computing" and "computation".

Word Embeddings

Several vector space models have been used in distributional semantics (Deerwester et al., 1990; Blei et al., 2003). Among these models, Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) represent words in a continuous subspace of a predefined number of dimensions. LDA is a topic model that generates topics based on word frequency from a set of documents (Blei et al., 2003). LSA uses singular-value decomposition, a technique closely related to eigenvector decomposition and factor analysis, to model the associative relationships between a set of documents and the terms they contain (Deerwester et al., 1990).

NNs have been successfully applied in diverse Natural Language Processing applications, such as language modeling (Bengio et al., 2003; Turian et al., 2010; Collobert et al., 2011; Mikolov et al., 2013), speech recognition or machine translation. Mikolov et al. (2013) developed two successful approaches with the so-called Continuous Bag-Of-Word (CBOW) and skip-gram models to build continuous word representations, i.e., word embeddings (Figure 2.2). The CBOW model attempts to predict the current word knowing its context. The skip-gram model aims at predicting a word, basing its decision on other words in the same sentence. It uses a window to limit the number of words used; e.g. for a window of 5, the system classifies the word w from the 5 words before and the 5 words after. Given a sequence of training words $w_1, w_2, w_3, ..., w_n$, the objective of the skip-gram model is to maximize the average log probability:

$$\frac{1}{N}\sum_{i=1}^{n}\sum_{-ws\leq j\leq ws, j\neq 0} \log prob(w_{i+j} \mid w_i) \tag{2.2}$$

where ws is the window size and n is the number of words in the training set.

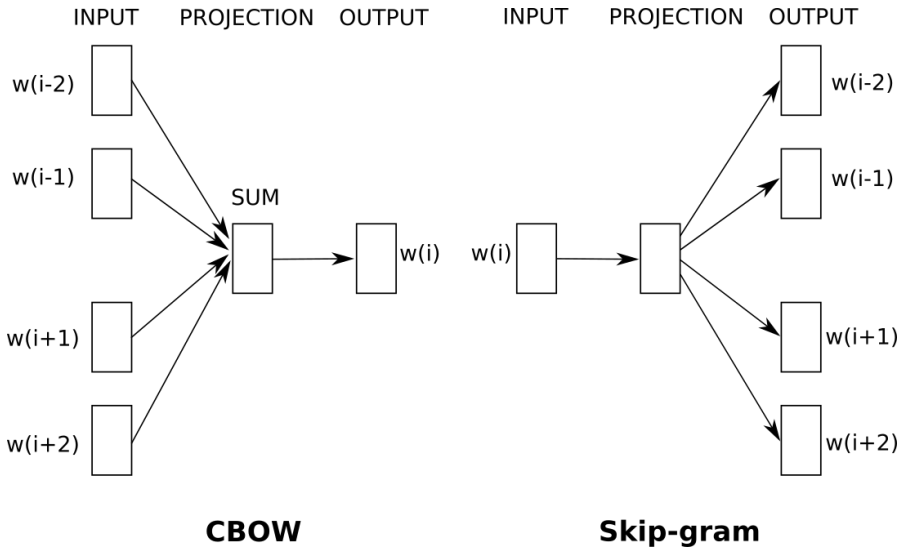


Figure 2.2: *Continuous Bag-of-Words and skip-gram models (Source: (Mikolov et al., 2013)).*

On the one hand, one-hot encoding uses binary vectors that have a dimensional size equal to the size of vocabulary of a document, i.e. this representation is different for each document and it does not represent a context for a word. On the other hand, word embeddings have a continuous and limited representation, i.e. vectors have the same dimension size for different documents and preserve the context of words (close words have similar contexts). Figure 2.3 illustrates the ability of the model to automatically organize concepts and to learn implicitly the relationships between them. Word embeddings can be used with different documents; however, some documents may contain out-of-vocabulary words because these words did not exist in the corpus used to learn the word embedding space. In this case, state-of-the-art works normally use a random or zero representation for these words. Recent works (Bojanowski et al., 2017a; Li et al., 2018) propose the analysis of subwords to minimize the out-of-vocabulary problem.

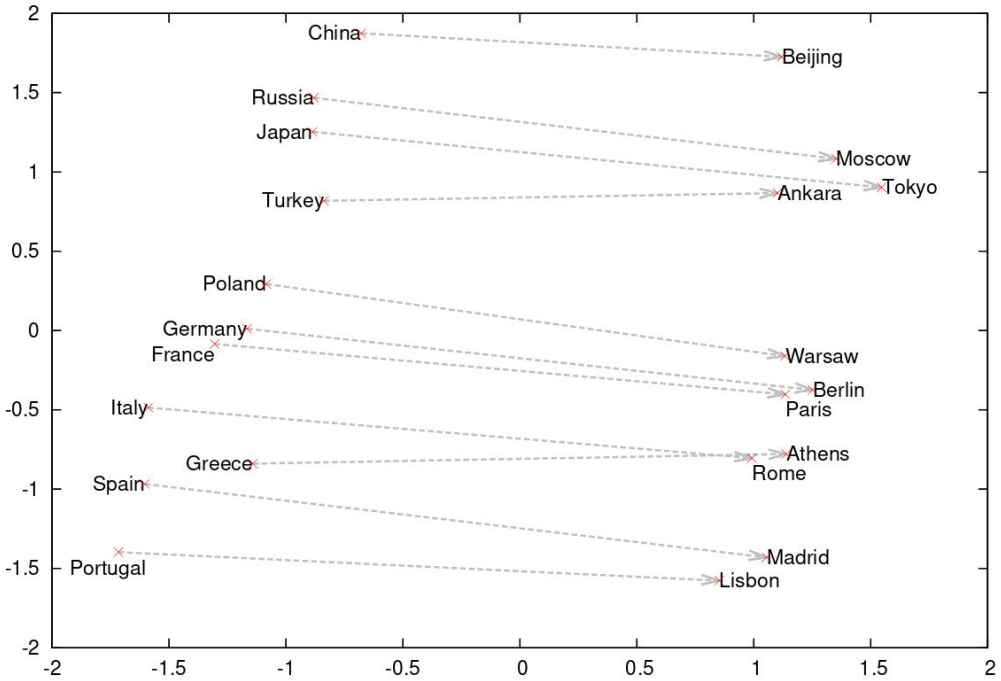


Figure 2.3: *Example of two-dimensional principal component analysis projection of the 1000-dimensional skip-gram vectors of countries and their capital cities (Source: (Mikolov et al., 2013)).*

Continuous vector representations generated by Mikolov (Mikolov et al., 2013) improved the word analysis because this representation gets the context of words and enables a better analysis of sentences, by improving the performance of several systems in the state of the art.

There are several other word embedding representations, the most notable ones being Glove (Pennington et al., 2014) and FastText (Bojanowski et al., 2017b). Pennington et al. (2014) proposed a specific weighted least squares model that trains on global word-word co-occurrence counts. They used a global log-bilinear regression model that combines global matrix factorization and local context window methods. Bojanowski et al. (2017b) proposed the FastText approach based on the skip-gram model, where

each word is represented as a bag of character n-grams. A vector representation is associated with each character n-gram, words being represented as the sum of these representations. Their method is fast, allowing models to be trained on large corpora quickly and to compute word representations for words that did not appear in the training data.

Word embeddings improved one-hot encoding by keeping the context of words at several levels. However, word embeddings need a large corpus and their representation depends on the vocabulary and the subjects covered in these texts. For example, word embeddings learned from medical and humoristic datasets may provide different contexts for the same words.

NNs are not only used to build word representations. Nowadays, recent systems in TS, MT and CLTS have used NNs as to improve their approaches. Therefore, the next section introduces the theory of NNs to understand how these systems work.

2.2 Neural Networks

Artificial NNs are inspired by the biological NN that constitute human brains. Human brains learn to solve complex problems (such as image recognition and/or language model) by considering examples. For instance, human brain learns to identify a house by seeing several types of houses every day until our brain can generalize how to identify a house. Human brains are composed of several neurons that are responsible for this learning process.

A biological neuron is an electrically excitable cell that receives, processes, and transmits information through electrical and chemical signals. It is composed of a cell body, dendrites, and an axon (Figure 2.4). Dendrites are thin structures that arise from the cell body which receives inputs from the other cells. The axon is the output structure of the neuron; when a neuron wants to communicate to another neuron, it sends an electrical message throughout the entire axon. Most neurons receive signals via the dendrites and send out signals down the axon. The connection of several neurons compose a neural network (Hertz et al., 1991).

Artificial NNss have functions and structures corresponding to biological NNs of animal and human brains. A biological neuron is represented by an artificial neuron named Perceptron (Figure 2.5). Dendrites are represented by inputs and weights, cell body by a transfer function and an activation function, and axon by the output of activation. The transfer function receives inputs, multiplies them by their weights and sums up their results. Finally, this artificial neuron applies a non-linear activation function (Section 2.2.2) in the output of transfer function and activates itself if its result exceeds a threshold (McCulloch and Pitts, 1943).

A Perceptron can solve linear problems such as AND and OR logical gates (Figure 2.6). However, XOR logical gate, which cannot be modeled through a linear separator, cannot be solved with a single artificial neuron ((Goodfellow et al., 2016), Chapter 6).

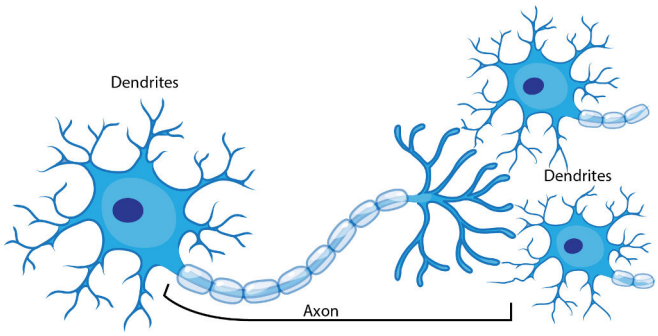


Figure 2.4: *Biological neurons in human brains (Source:* `https://askabiologist.asu.edu`*).*

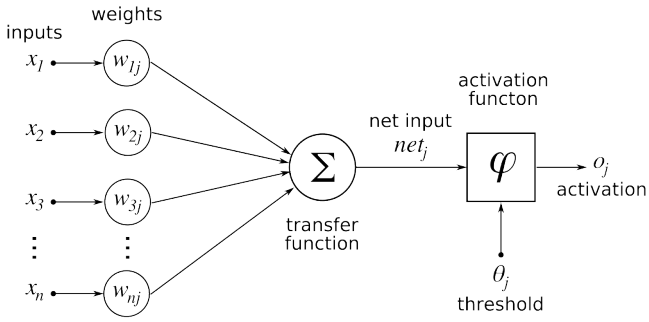


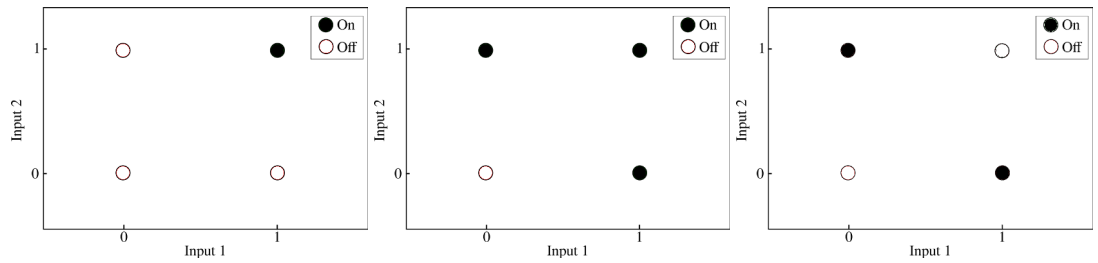
Figure 2.5: *Artificial neuron (Source:* `https://www.kdnuggets.com`*).*

NNs combine several artificial neurons in several layers to solve more complex problems, e.g. the XOR gate, image recognition, language model and so on.

The following subsections describe artificial NNs in more details. Section 2.2.1 presents the most relevant NN architectures for TS and MT applications. Section 2.2.2 analyzes several activation functions used in NN. Sections 2.2.3 and 2.2.4 explain the back-propagation algorithm and some regularization methods to generalize data and to avoid overfitting problems, respectively. Finally, attention mechanisms are described in Section 2.2.5.

2.2.1 Architectures

NNs are grouped according to their neuron arrangements. There are several NN architectures; in this thesis, we only describe the NN architectures most related to our work: FeedForward Neural Networks, AutoEncoders, Convolutional Neural Networks and Recurrent Neural Networks.



(a) AND. (b) OR. (c) XOR.

Figure 2.6: *Logical gates.*

FeedForward Neural Network

FeedForward Neural Network (FFNN) is a combination of different numbers of artificial neurons by layer. A deep FFNN is composed of several hidden layers. FeedForward Neural Networks (FFNNs) are characterized by the flow of signals in only one direction. Input signals are fed into the input layer, then, after being processed, they are forwarded to the next layer, just as shown in Figure 2.7 (Rosenblatt, 1962). The size of NN models is determined by the number of neurons and layers. Large NN models increase the complexity and the capacity of NNs to solve more complex problems. However, these models are more difficult to train (Sections 2.2.3 and 2.2.4).

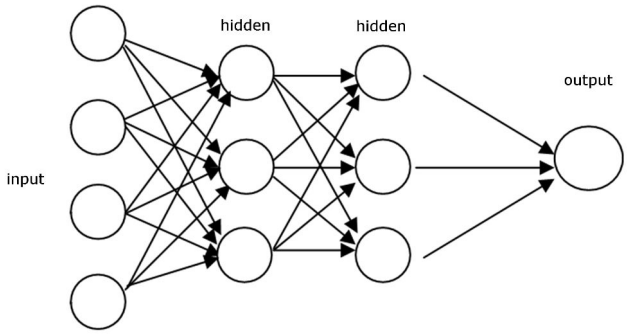


Figure 2.7: *An example of a deep FeedForward Neural Network with 2 hidden layers.*

AutoEncoder

AutoEncoder (AE) is a special kind of FFNN which aims to reproduce some input data using a smaller representation. The input layer and the target output are typically the same. Hidden layers decrease and increase the representation of information. The bottleneck layer is a hidden layer with a reduced dimension. The left side of this bottleneck layer is an encoder and the right side is a decoder (Figure 2.8). An encoder typically reduces the dimension of the data and a decoder increases the dimensions (Liou et al.,

2014; Shanmugamani, 2018). Recent methods also use RNNs and Convolutional Neural Network (CNN) to represent the encoder and the decoder.

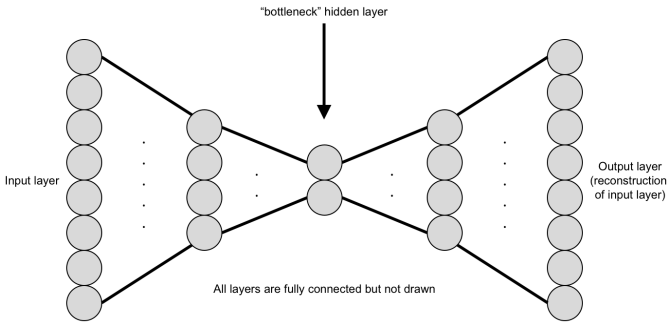


Figure 2.8: *An example of AutoEncoder using FFNN (Source: (Shanmugamani, 2018)).*

Convolutional Neural Network

Inspired by the structure of mammals' visual cortexes, CNN is a kind of FFNN characterized by convolutional, pooling and fully-connected layers (LeCun and Bengio, 1998; Albelwi and Mahmood, 2017). Originally invented for computer vision, an input window slides along the image. The data are passed to convolution and pooling layers to extract local features and to remove non-relevant data, respectively (Figure 2.9). Beyond analyzing images, CNNs are also used in machine translation, caption generation and several other NLP applications.

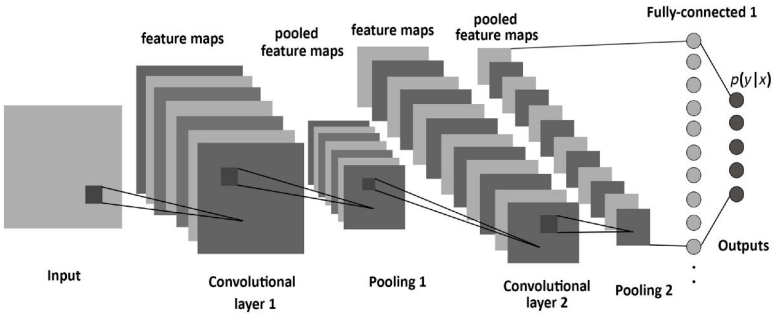


Figure 2.9: *An example of Convolutional Neural Network to process and to classify an image among several classes (Source: (Albelwi and Mahmood, 2017)).*

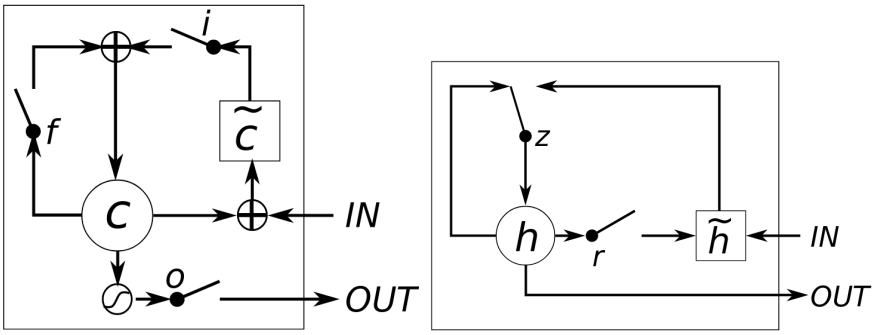
Recurrent Neural Network

Recurrent Neural Network (RNN) is a class of NNs where connections between nodes form a directed graph along a sequence. These connections allow RNNs to exhibit dynamic temporal behavior for a time sequence. Unlike FFNNs, RNNs can use their internal state (memory) to process sequences of inputs. This type of NNs is mainly

used when context is important or when decisions from past iterations or samples can influence current ones.

Long Short Term Memory (LSTM) is a recurrent cell that is composed of a memory to preserve the information and several gates that decide whether to pass the data forward, erase memory and so on (Figure 2.10(a)). The input gate decides how much information from the last sample will be kept in memory; the output gate regulates the amount of data passed to the next layer, and the forget gate controls the tearing rate of memory stored (Chung et al., 2014; Greff et al., 2015).

Gated Recurrent Units (GRUs) are a simpler version of LSTM. They are composed of recurrent cells that only have an update gate and a reset gate (Figure 2.10(b)). The update gate retains relevant information and the reset gate chooses when recurrent cells process the input information.



(a) LSTM. (b) GRU.

Figure 2.10: *Illustration of (a) LSTM and (b) GRU. (a) i, f and o are the input, forget and output gates, respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation (Source: (Chung et al., 2014)).*

2.2.2 Activation Functions

Activation functions give different characteristics to NNs. Each activation function has a specific behavior that changes how a NN learns a task. Some activation functions are more complex than others. On the one hand, complex functions can increase the complexity and the generalization of NNs. On the other hand, they increase the time of learning process by complicating the convergence of NNs.

Among several activation functions, linear, ReLU, sigmoid and hyperbolic are the most popular functions (Table 2.1 and Figure 2.11).

Activation Function	Equation	Derivative
Linear	$f(x) = x$	$f'(x) = 1$
ReLU	$f(x) = \max(0, x)$	$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = \sigma(x)(1-\sigma(x))$
Hyperbolic	$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'(x) = 1 - \tanh(x)^2$

Table 2.1: *Activation functions and their equations.*

Linear

The linear function generates an output proportional to the input, i.e. huge inputs generate huge output values (Hertz et al., 1991). The gradient of this function is a constant, therefore the gradient is independent of the input data.

ReLU

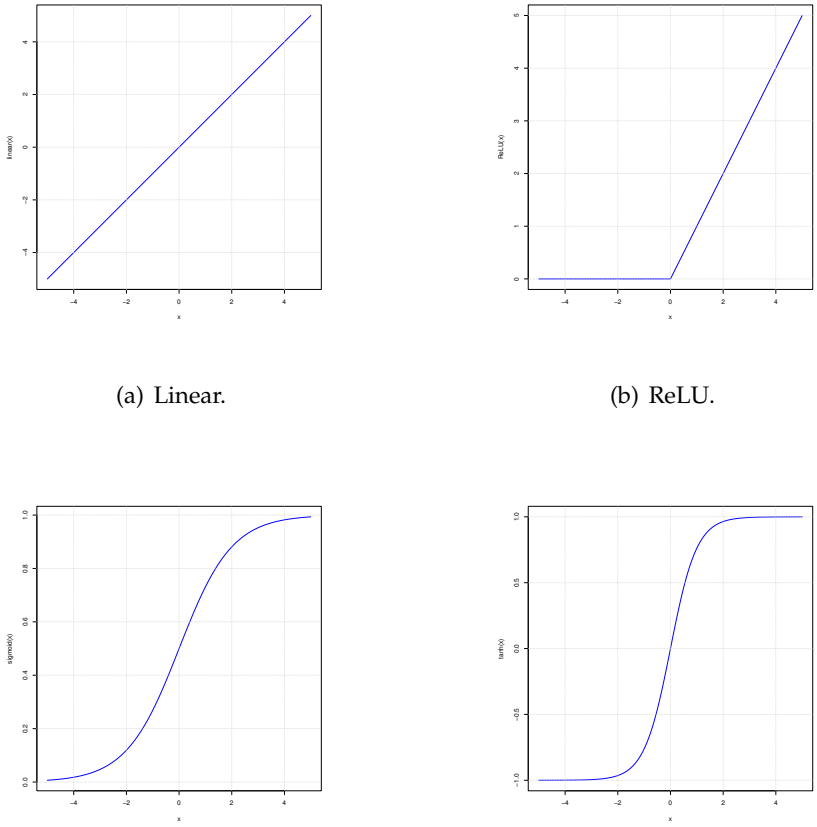
Finally, Rectified Linear Unit (ReLU) is another nonlinear function with the output range [0, ∞) (Hahnloser et al., 2000; Nair and Hinton, 2010). This function can generate huge output values and the gradient can go towards 0, which stops the learning process because it gives the output value 0 for negative X. ReLU has several variations (Leaky ReLU, parametric ReLU, and randomized leaky ReLU) to mitigate these problems by simply making the horizontal line into non-horizontal components. This function is popular because it is less computationally expensive than sigmoid and hyperbolic functions.

Sigmoid

The sigmoid is a nonlinear function that has an output range (0,1) (Nair and Hinton, 2010). This value range avoids the generation of huge output values, which gives stability to NNs. However, the gradient of this function is smaller and may become very small in NNs with several layers. This process is known as "vanish gradients" and it happens when loss error disappears in the back-propagation process, prejudicing the learning process of NNs.

Hyperbolic

The hyperbolic function (tanh) is also a nonlinear function and has an output range bigger than the sigmoid function (-1,1) (Hertz et al., 1991). This function has charac-



(a) Linear. (b) ReLU.

(c) Sigmoid (d) Hyperbolic.

Figure 2.11: *Plot of activation functions.*

teristics similar to the sigmoid. The bigger the output range the higher the gradient of this function, thus helping the learning process. However, tanh also has the vanishing gradient problem.

2.2.3 Learning Process

The learning process consists in setting up the parameters of NNs to process input data to generate correct outputs. The back-propagation algorithm minimizes the loss function that measures the difference between expected and calculated outputs. This algorithm uses the output error and the gradient of functions to update the values of parameters in order to minimize the output error, i.e. it propagates the error backwards by updating the values of parameters. Since this method requires the computation of the gradient of functions at each iteration step, activations and the loss functions have to be continuous and differentiable (Goodfellow et al., 2016).

2.2.4 Regularization

NNs attempt the generalization of input data to generate good answers with new data. NNs overfit when they achieve good performance on the training data and poor results on unseen data (Figure 2.12). This problem happens when systems memorize the training data because of small amount of training data and/or neural network architecture problems.

The most used methods to mitigate this problem are L1 and L2 regularizations, and dropout. These methods generate different values for the loss function by avoiding NNs memorize results for some specific input data and helping NNs learn how to generalize the input data ((Goodfellow et al., 2016), Chapter 7).

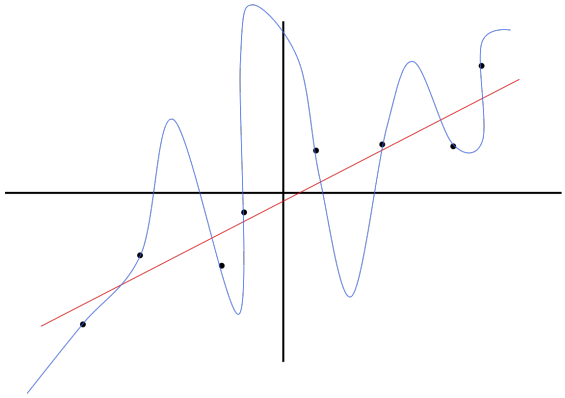


Figure 2.12: *An overfitting example: the red line generalizes better the behavior of points than the blue line (Source:* `http://nikhilbuduma.com/2014/12/29/deep-learning-in-a-nutshell/`*).*

L1 and L2 Regularizations

L1 and L2 regularizations consist in adding a noise to avoid the generation of the same results for given input data. This noise is generated by adding the weights (**W**) of NN to the result of the loss function (*loss*). On the one hand, L1 regularization (Equation 2.3) penalizes small weights, and tends to concentrate the weights of NNs on a small number of connections. On the other hand, L2 regularization (Equation 2.4) penalizes large weights, and tends to make the network have small weights ((Goodfellow et al., 2016), Chapter 7).

$$L1 = loss + \frac{\lambda}{n} \sum_{i}^{nl} \sum_{j}^{na_i} |W_{ij}| \tag{2.3}$$

$$L2 = loss + \frac{\lambda}{n} \sum_{i}^{nl} \sum_{j}^{na_i} (W_{ij})^2 \tag{2.4}$$

where n is the number of weights in the NN, nl is the number of layers and na_i is the number of artificial neurons in the layer i.

Dropout

Unlike L1 and L2 regularizations, dropout modifies the network architecture. Dropout removes nodes of NNs at random (Figure 2.13). This process can be applied in one or several layers in a NN. During the training process, the dropout method generates different NNs. The same input data generate several loss function outputs (one for each NN configuration), reducing the memorization problem. This methodology helps the NN to generate the same output with several kinds of neurons arrangements (Srivastava et al., 2014; Goodfellow et al., 2016).

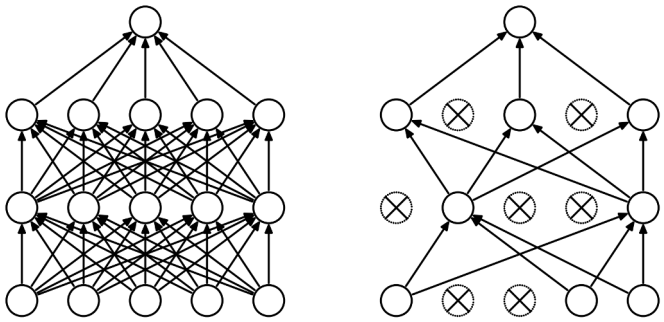


Figure 2.13: *A dropout example (Source: (Srivastava et al., 2014)).*

2.2.5 Attention Mechanism

Attention mechanisms improved the performance of several systems in the state of the art (Bahdanau et al., 2014). They are based on the visual attention mechanism found in humans and they are mainly used in the decoding process of NNs. Instead of analyzing all input data, the attention mechanism helps the decoder to focus on different parts of the input data at each step of the decoding process. During the learning process, the NN learns which part of the input data to focus on, i.e. what to attend to, based on the input data and what it has generated so far.

Among several kinds of attention mechanisms in the state of the art (Bahdanau et al., 2014; Xu et al., 2015; See et al., 2017; Vaswani et al., 2017), the following two subsections describe the two most studied attention mechanisms (soft and hard attention mechanisms).

Soft Attention

Soft attention analyzes the average relevance of inputs to generate a new output at each time step. Bahdanau et al. (2014) proposed a conditional probability to generate

a solution considering the relevance of RNN input at each time step. The generation of outputs depends on a context $\mathbf{c}_i$ for each target output y_i. The context vector $\mathbf{c}_i$ depends on a sequence of annotations ($\mathbf{h}_1$, ..., $\mathbf{h}_T$) to which an encoder maps the input data (Figure 2.14):

$$p(y_i|y_1, ..., y_{i-1}, x) = g(y_{i-1}, \mathbf{hs}_i, \mathbf{c}_i), \tag{2.5}$$

where g is a nonlinear function and $\mathbf{hs}_i$ is an RNN hidden state for time i, computed by

$$\mathbf{hs}_i = f(\mathbf{hs}_{i-1}, y_{i-1}, \mathbf{c}_i). \tag{2.6}$$

Each annotation $\mathbf{h}_i$ contains information about the input data with a strong focus on the parts surrounding the i-th input. The context vector $\mathbf{c}_i$ is computed as a weighted sum of these annotations:

$$\mathbf{c}_i = \sum_{j=1}^{T} \alpha_{ij} \mathbf{h}_j \tag{2.7}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})} \tag{2.8}$$

$$e_{ij} = a(\mathbf{hs}_{i-1}, \mathbf{h}_j) \tag{2.9}$$

The alignment model a is parametrized as a FFNN which is jointly trained with all the other components of the NN, where e_{ij} determines the relevance of the input j to the output i. The attention model is known as soft attention because it allows the gradient of the cost function to be backpropagated through.

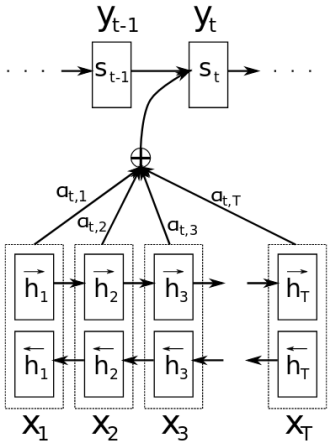


Figure 2.14: *Soft-attention mechanism in sequence-to-sequence model (Source: (Bahdanau et al., 2014)).*

Hard Attention

Hard attention replaces the deterministic method of soft attention with a stochastic sampling model. Instead of calculating a weighted average of the input, hard attention uses a sample rate to pick one area of the input. As the sample rate is not derivable, the back-propagation cannot be applicable. Instead, the Monte Carlo method performs several end-to-end episodes to compute an average for all sampling results to execute the back-propagation method (Xu et al., 2015).

The release of large datasets and the advances in NNs have improved the results of several NLP applications, e.g. sentence similarity (Chapter 3), TS (Section 2.3), MT (Section 2.5) and so on. However, NNs have poor performance for tasks with small datasets. Recent studies have used transfer learning techniques to minimize this problem of low resources for some languages and/or tasks (Mou et al., 2016). The next section focuses on the Mono-language Text Summarization using several approaches (optimization, heuristics, NNs and so on) and different word representations.

2.3 Mono-Language Text Summarization

Text Summarization (TS) aims to generate a short, correct and informative summary that describes the main information of one or several documents. TS systems can be ranged in various types according to the following perspectives (Torres-Moreno, 2014):

- Summary generation can be extractive, compressive or abstractive. Extractive methods estimate the relevance of sentences in a document to generate a summary by concatenating the most relevant sentences. Compressive methods compress sentences to reduce the length of sentences and to preserve only the main information. Then, they generate summaries by concatenating the most relevant sentences and compressions of a document. Finally, abstractive methods analyze a document and generate a summary with new sentences that contain the meaning of the source documents.

- Summaries can be generic, contractive or focused on a specific topic. Generic abstracts do not differentiate the content addressed in the documents and perform the same type of analysis on all texts (Gong and Liu, 2001). Contrastive summarization jointly generates summaries for two entities in order to highlight their differences. Those that are focused on a topic have more advanced rules and concepts for an area of analysis. For example, the determination of the relevance of criminal cases by analyzing their summaries (Moens et al., 2004).

- The creation of summaries can be mono or multi-documents. Mono-document summarization analyzes a text and creates the summary with its main information. Multi-document summarization usually analyzes a cluster of documents that usually contains similar information.

- Summarization can be mono-language, multi-language or cross-language. Monolingual summarization generates a summary in the language of source documents. Multi-lingual summarization analyzes source documents in several languages to produce a summary in one of the languages presented in the source documents. Cross-language summarization aims to generate a summary of a document where the summary language differs from the document language.

Among several campaigns to evaluate NLP algorithms, DUC/ TAC[2] has organized workshops to analyze and to compare the performance of TS systems. This campaign provides datasets that allow participants to analyze their systems by comparing them to other state-of-the-art systems. The following subsections describe the most relevant works split in extractive, compressive and abstractive methods.

2.3.1 Extractive Methods

Early work on automatic document summarization addressed only journalistic texts using simple techniques based on frequency of words to evaluate the relevance of sentences (Luhn, 1958). On the one hand, summaries of professionals have great quality in terms of information and readability. Their production is slower, more expensive and subject to the subjectivity of the professional. On the other hand, abstracts produced automatically have a very low cost of production, while they are not prone to subjectivity and variability problems observed in the propositions of professional abstracter, among others. There was thus a need to generate automatic summaries to deal with the growth of the amount of information. Edmundson (1969) gave continuity to Luhn's works, by adding to the process of producing summaries considerations on the position of sentences and the presence of words from the document structure (e.g. titles, sub-titles, etc.).

Since this pioneering work, new methods have been published to improve TS using heuristic methods, graph theory, Integer Linear Programming (ILP) methods, submodular functions, NNs and so on. As extractive methods are composed of a large number of works, we only present a few representative works in the following subsections.

Heuristic Methods

Gong and Liu (2001) proposed a generic extractive text summarization method to identify semantically important sentences for generating the summary. Their approach uses the latent semantic analysis technique to calculate the relevance of sentences. Unlike the generic summaries of Gong and Liu (2001), Moens et al. (2004) generate criminal case summaries. They presented the SALOMON project that uses knowledge bases to identify the relevant information units of the legal texts. Then, their system uses statistical techniques to extract informative text units of the alleged offenses and of the opinion of the court.

[2] `http://duc.nist.gov` and `http://www.nist.gov/tac`

Boudin and Torres Moreno (2007) proposed a multilingual single-document summarization system based on the vector-space model. This system combines several statistical processing operations (calculating entropy, frequential weight of segments and words, Hamming measures, etc.) with a decision algorithm. It consists of four stages: statistical language identification, preprocessing and vectorization; the computation of the metrics; a decision algorithm combining the information of the metrics; and the summary generation with a simple post-processing.

The work of (Xu et al., 2013) uses the concepts of hierarchical topical tree, rhetoric, and temporal relation to calculate the interrelationships between the units of the text. It also considers the multi-document rhetorical structure to represent a text at different levels of granularity (including sentences, paragraphs, sections, and documents). Its extraction algorithm performs steps of weighting and removal of nodes in order to select the most important sentences. First, it performs the node weighting algorithm; then it uses a clustering algorithm in order to identify similar sentences and remove those that are redundant.

Graph Theory

A graph is an ordered pair $G = (V, E)$ comprising a set V of vertices or nodes or points together with a set of edges E or arcs or lines, which are 2-element subsets of V. In TS, nodes and edges represent the sentences of a document and the similarity between them, respectively. Sentence weighting methods use the overall information in the graph to calculate the importance of each sentence.

Mihalcea and Tarau (2004) proposed a successful graph-based approach that was originally devised to estimate the relevance of pages from the number of citations or the study of the Web structure. This system makes decisions about the importance of a vertex based on the global information coming from the recursive analysis of the complete graph. In the scope of automatic summarization, it is observed that the document is represented by a graph of textual units (sentences) connected to each other through relations resulting from similarity calculations. The sentences are then selected according to the criteria of centrality or prestige in the graph, and grouped in order to produce extracts of the text (Ferreira et al., 2014). Similarly to (Mihalcea and Tarau, 2004), the LexRank method calculates the sentence importance based on the concept of eigenvector centrality in a graph representation of sentences (Erkan and Radev, 2004).

Baralis et al. (2013) also used graphical modeling to summarize texts. Their methodology is composed of: text processing, graph correlation, graph indexing and sentence selection. The text processing step performs the stemming and the removal of the stopwords. The correlation of the graph is made from the sets of items that are frequent in the text. The indexing of the graph occurs through an algorithm based on the PageRank to weight the nodes of the graph. Finally, the summary generation selects the best weighted sentences based on node indexing.

Linhares Pontes et al. (2014) combined the Graph Theory with the Jensen-Shannon divergence to create multi-document summaries by extraction. This method represents

the text through a graph where the sentences correspond to the vertices and the edges represent the similarity between them. Therefore, the group of vertices interconnected in the graph represent sentences with similar content. Then, the maximum stable set of the graph is composed to create the summary with sentences containing the general information of the cluster and without redundancy. Therefore, the system usually selects a sentence from each group to reduce the redundancy of summaries.

Fang et al. (2017) proposed a word-sentence co-ranking model which combines the word-sentence relationship with the graph-based unsupervised ranking model. In this analysis, the mutual influence is able to convey the intrinsic status of words and sentences more accurately.

ILP Methods

One way to improve the quality of summaries is to maximize the selection of the most relevant sentences. Integer Linear Programming (ILP) can be used to model text summarization in order to maximize the quality of sentence extraction through text analysis. In ILP, problems are seen as maximizing or minimizing an objective function to a set of constraints.

McDonald (2007) considered the informativeness and redundancy of sentences as key points for TS. He evaluates the quality of a summary from the relevance of sentences with the insertion of a penalty to the redundant sentences. Gillick and Favre (2009) relied on the Mcdonald's model to address the shape scale model. The authors treat the redundancy of sentences without requiring a quadratic number of variables, thus facilitating the modeling and resolution of the problem. They also model this problem as an ILP formulation by performing the compression and the selection of sentences simultaneously.

Combining regression and ILP methods, Galanis et al. (2012) used a Support Vector Regression (SVR) model to measure the relevance of the sentences from the training of systems with summaries produced by humans. This model evaluates the relevance of sentences based on their diversity. Oliveira et al. (2017) generated multiple candidate summaries for each input article by exploring different concept weighting methods and representation forms using an ILP method. Then, a regression model enriched with several extracted features at the levels of summary, sentence and n-gram level is trained to select the most informative summary based on an estimation of the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score.

Submodular Function

Submodular functions share a number of properties in common with convex and concave functions (Lovász, 1983), including their wide applicability, their generality, their multiple options for their representation, and their closure under a number of common operators. Therefore, submodular function-based approaches are an alternative to the

ILP approaches providing good results without the complexity of ILP models. Let S be a finite set of sentences, the modular function f satisfies, for any $A \subseteq B \subseteq S \setminus s$, the following property:

$$f(A + \{s\}) - f(A) \geq f(B + \{s\}) - f(B) \tag{2.10}$$

where $s \in S$. This property shows the intuition that adding a sentence to a small set of sentences (i.e., summary) makes a greater contribution than adding a sentence to a larger set. Therefore, TS by extraction can be represented by the selection of a subset of sentences Sum to represent a document S ($Sum \subseteq S$) in such a way that the summary contains the most relevant sentences of the document with a limited size.

Lin and Bilmes (2011) designed a class of submodular functions to generate summaries aiming at the diversity and representativeness of the corpus. The functions are non-decreasing, monotonous and submodular, allowing an ideal constant factor performance. Then, Dasgupta et al. (2013) generalized the submodular framework of (Lin and Bilmes, 2011). Their framework is composed of a submodular function and a non-submodular dispersion function. This dispersion uses inter-sentence dissimilarities in different ways in order to ensure non-redundancy of the summary.

Another possibility is the supervised learning of submodular functions for the extraction of sentences. Sipos et al. (2012) applied this learning method to several submodular compaction methods and demonstrated their effectiveness based on the analysis of several datasets.

Neural Networks

Recent methods have used word embeddings and NNs to improve the sentence analysis and the selection of relevant sentences by generating more informative summaries: Kågebäck et al. (2014) and Yin and Pei (2015) with FFNN, and Cao et al. (2015) and Nallapati et al. (2017) with RNN.

Kågebäck et al. (2014) proposed the use of continuous vector representations for semantically aware representations of sentences as a basis for measuring similarity using submodular functions to select the most informative sentences. Unlike Kågebäck et al. (2014) that used vector addition and Recursive AutoEncoder (RAE) to generate a continuous representation of sentences, Yin and Pei (2015) proposed the Convolutional Neural Network Language Model (CNNLM) based on CNNs to project sentences into dense distributed representations. Then, they generated a summary composed of sentences that had high prestige and dissimilarity between them.

Cao et al. (2015) ranked sentences for Multi-Document Summarization (MDS) using RNN. They formulate the sentence ranking task as a hierarchical regression process, which simultaneously measures the salience of a sentence and its constituents (e.g., phrases) in the parsing tree. Ranking scores of sentences and words are utilized to effectively select informative and non-redundant sentences to generate summaries.

Nallapati et al. (2017) proposed a NN model composed of two-layer RNN-based sequence classifier: the bottom layer operates at word level within each sentence and the top layer decides if a sentence remains in the summary. The decision at each sentence depends on its content richness, its salience with respect to the document, its novelty with respect to the accumulated summary representation and other positional features.

Yousefi-Azar and Hamey (2017) introduced a stochastic version of an AE that adds noise to the input text to select the top sentences from a text. They evaluated how AEs handle a sparse word representation such as Term Frequency-Inverse Document Frequency (TF-IDF) and a less sparse word representation based on a document-specific vocabulary.

In spite of the recent advances in extractive summarization using optimization and NNs, sentence extraction generates summaries by selecting the most relevant sentences from the source document. This selection of sentences assures the grammaticality and the concision at local (sentence) level but not at global (summary) level. Another limitation is that these summaries keep irrelevant words of extracted sentences. The compression and the generation of sentences can generate new sentences containing only the main information, making the summary more concise.

2.3.2 Compressive Methods

In recent years, some progress has been made to go beyond extractive summarization, especially in the context of compressive summarization. Several works used ILP model to compress and to summarize texts (Martins and Smith, 2009; Li et al., 2013, 2014; Yao et al., 2015a).

Several TS approaches jointly perform sentence extraction and compression using ILP models (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013). Martins and Smith (2009)'s formulation combines dependency parsing information, which only requires a linear number of variables, with a bigram model. Berg-Kirkpatrick et al. (2011) scored candidate summaries according to a combined linear model whose features factor over the n-gram types in the summary and the compressions used. Then, they jointly formulated sentence selection and syntax tree trimming in integer linear programs. Almeida and Martins (2013) developed a dual decomposition framework for compressive TS. Their models for sentence compression and extractive summarization were trained by multi-task learning techniques.

Jorge et al. (2010) proposed a compression method based on the cross-document structure theory model that analyzes redundancy, complementarity and contradiction between the different sources of information to calculate the relevance of sentences. These metrics are used in the process of reweighting sentences in order to create summaries. Woodsend and Lapata (2012) proposed a model that attempts to cover content selection, surface realization, paraphrasing, and stylistic conventions. These aspects are learned separately and are jointly optimized using an ILP model to generate the output summary. Wang et al. (2013) compared several sentence compression techniques for query-focused MDS. They designed three types of approaches:

sentence-compression-rule-based, sequence-based and tree-based and examine them within their compression-based framework for query-specific MDS. Their tree-based method compresses sentences by removing non-relevant branches of their parse tree based on query relevance, content importance, redundancy and language quality, among other measures.

Li et al. (2013) proposed summaries guided by a compression method combined with an ILP-based sentence selection. They trained a supervised sentence compression model using a set of word-, syntax-, and document-level features. During summarization, they used multiple compressed sentences in the ILP framework to select salient summary sentences. Then, Li et al. (2014) introduced a sentence compression model based on expanded constituent parse trees. Their model used an expanded constituent parse tree to extract rich features for every node in the constituent parser tree. They introduced a pipeline summarization framework where multiple compression candidates were generated for each pre-selected important sentence, and then an ILP-based summarization model was used to select the final compressed sentences.

Qian and Liu (2013) proposed an efficient decoding algorithm for fast compressive summarization using graph cuts. Their approach first relaxed the length constraint using Lagrangian relaxation. Then they proposed to bound the relaxed objective function by the supermodular binary quadratic programming problem, which can be solved efficiently using graph max-flow/min-cut. Yao et al. (2015a) formulated a sparse optimization framework for compressive document summarization. They introduced an additional sentence dissimilarity term to encourage diversity in summary sentences. The resulting sparse optimization problem is jointly non-convex, so they derived a block coordinate descent algorithm to solve it, followed by a recursive sentence compression phase to impose grammatical constraints.

Several works (Banerjee et al., 2015; Sun et al., 2015; Niu et al., 2017; Yao et al., 2015a; Nayeem et al., 2018) used MSC methods to compress sentences and to improve the informativeness of TS systems. Banerjee et al. (2015) developed a multi-document TS system that generated summaries based on compressions of similar sentences. They used the Filippova (2010)'s method to generate 200 random compressed sentences (more details in Chapter 4). Then they created an ILP model to select the most informative and grammatically correct compression. Sun et al. (2015) proposed an event-driven model for headline generation. Their system identifies a key event chain of a document by extracting a set of structural events that describe them. Then a MSC algorithm is used to fuse the extracted events, by generating a headline for the document. Niu et al. (2017) proposed a compressive MDS system based on Chunk-Graph (CG) and Recurrent Neural Network Language Model (RNNLM). In their approach, a CG based on word-graph was constructed to organize all information in a sentence cluster. They used beam search and character-level RNNLM to generate readable and informative summaries from the CG for each sentence cluster. Recently, Nayeem et al. (2018) designed an abstractive sentence generation model which jointly performs sentence fusion and paraphrasing using skip-gram word embedding model. Their sentence generation model combined the word graph model and lexical substitution of words to generate abstractive compressions without losing grammaticality. They jointly used

this sentence generation model and ILP model to generate abstractive summaries.

On the one hand, compressive methods attempt to remove irrelevant information of sentences in order to retain only the main information. This type of approach allows the generation of summaries shorter than extractive methods without reducing informativeness. On the other hand, extractive methods produces more correct summaries than compressive methods. While extractive approaches generate summaries by reusing extracted sentences which ensures grammaticality at local level, compressive methods attempt to produce new shorter sentences which may have some grammatical mistakes at sentence and summary levels.

2.3.3 Abstractive Methods

Abstractive models generate summaries from scratch without being constrained to reuse sentences from the original text. Saggion and Lapalme (2002) presented the SumUM system to produce indicative informative summaries based on abstracts written by professional abstractors. This kind of summary provides the topics of the document, and the informative part elaborates on some of these topics according to the reader's interest. The SumUM system generates the summary based on a pre-established conceptual order, the merging of some types of information, and the reformulation of the information in one text paragraph. Genest and Lapalme (2012) introduced a full abstraction approach in the context of guided summarization. They used a rule-based, custom-designed information extraction module, integrated with content selection and generation in order to write short abstractive summaries. They designed specific rules and patterns to address a theme or subcategory of source documents.

Bing et al. (2015) proposed an abstraction-based MDS framework that can construct new sentences by exploring more fine-grained syntactic units than sentences, namely noun/verb phrases. Their method first constructs a pool of concepts and facts represented by phrases from the input documents. Then new sentences were generated by selecting and merging informative phrases to maximize the salience of phrases and meanwhile satisfy the sentence construction constraints. They employed integer linear optimization for conducting phrase selection and merging simultaneously in order to achieve the global optimal solution for a summary.

With the emergence of deep learning as a viable alternative for many NLP tasks, researchers have started considering this framework as an attractive, fully data-driven alternative to abstractive summarization. Rush et al. (2015) proposed an attentional feed-forward network for abstractive summarization of sentences into short headlines. Their model combines a neural language model with a contextual input encoder. Their method utilizes a local attention-based model that generates each word of the summary conditioned on the input sentence. Chopra et al. (2016) extended the Rush et al. (2015)'s work by proposing a conditional RNN model to generate abstractive summaries. The conditioning is provided by a convolutional attention-based encoder which ensures that the decoder using RNN focuses on the appropriate input words at each step of generation.

Several works analyzed TS with a sequence-to-sequence model where the input text is processed by an encoder and the summary is generated by a decoder (Hu et al., 2015; Cheng and Lapata, 2016). Hu et al. (2015) developed a sequence-to-sequence model using RNNs to summarize Chinese texts.

Due to limitations of sequence-to-sequence models that reproduce factual details inaccurately and often include repetitive and incoherent sentences for long documents, recent works extended them with attention mechanisms (pointer networks Nallapati et al. (2016), coverage (See et al., 2017), intra-temporal attention mechanism (Paulus et al., 2017)), the combination of generative and discriminative models (Liu et al., 2018), convolutional sequence-to-sequence (Wang et al., 2018), and deep recurrent generative decoder (Li et al., 2017) to generate more coherent and informative summaries.

Nallapati et al. (2016) extended Hu et al. (2015)'s work proposing an attentional encoder-decoder RNN to generate abstractive summaries. They used an hierarchical encoder with attention weights at the word level and at sentence-level attention weights. Their generation process contains a switch mechanism that selects a word from source sentences or from large vocabulary. This mechanism can generate summaries with unknown words by reusing words from source sentences. Then, See et al. (2017) proposed a hybrid pointer-generator network which helps accurate reproduction of information, while retaining the ability to produce novel words through the generator (Figure 2.15). They also applied a coverage analysis to keep track of what has been summarized and to avoid the generation of repeated words.

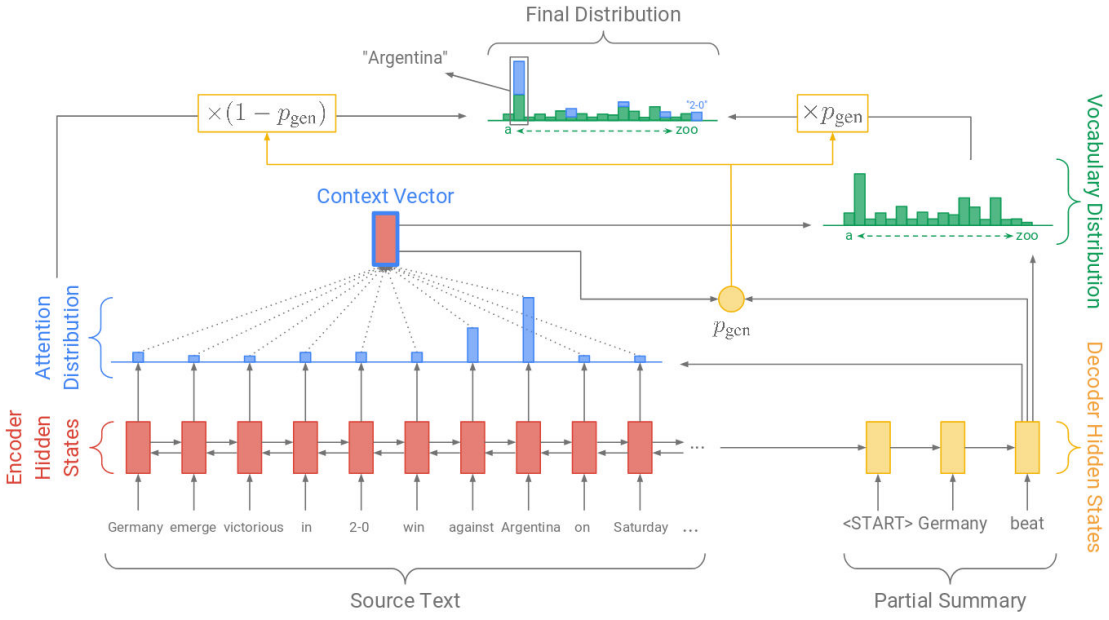


Figure 2.15: *Pointer-generator model. The probability $p_{gen} \in [0,1]$ is calculated at each decoder time step to generate a word from the vocabulary or copying a word from the source text (Source: (See et al., 2017)).*

Cheng and Lapata (2016) developed different classes of summarization models us-

ing a hierarchical document encoder and an attention-based extractor. Their NN model combines a CNN sentence encoder with a RNN to encode a text and the decoder can either select which sentences remain in the extractive summarization or extract words to generate abstractive summaries (Figure 2.16).

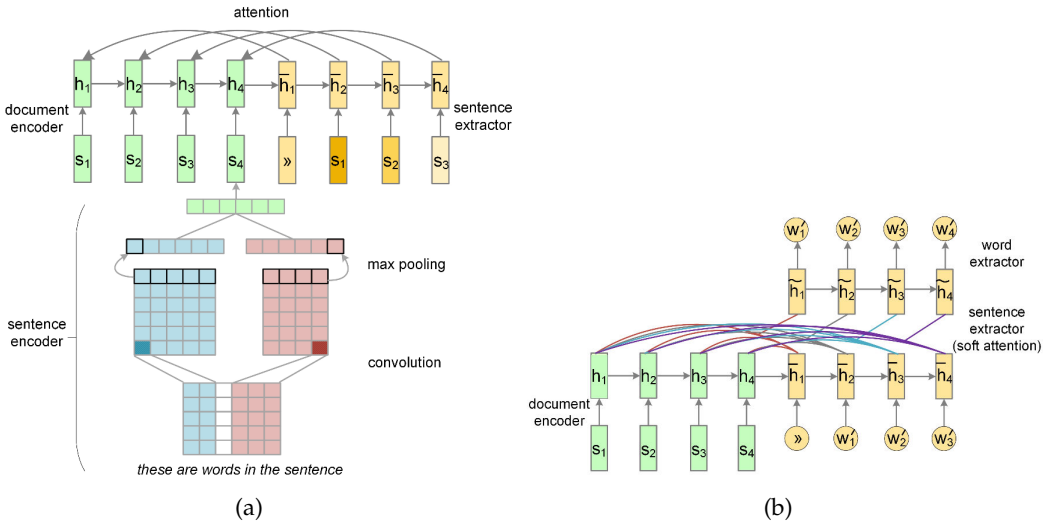


Figure 2.16: *Neural Network models for extractive (a) and abstractive (b) summarization (Source: (Cheng and Lapata, 2016)).*

Paulus et al. (2017) used an intra-temporal attention in the encoder that records previous attention weights for each of the input tokens while a sequential intra-attention model in the decoder takes into account which words have already been generated by the decoder to address the repeating phrase problem. Wang et al. (2018) proposed a joint attention and biased probability generation mechanism to incorporate the topic information. They employed the self-critical sequence training technique in convolutional sequence-to-sequence to directly optimize the model with respect to the non-differentiable summarization metrics ROUGE.

Li et al. (2017) extended the sequence-to-sequence model with a deep recurrent generative decoder to model and learn the latent structure information implied in the target summaries. They jointly considered the generative latent structural information and the discriminative deterministic variables to generate summaries. Liu et al. (2018) built a generator as an agent of reinforcement learning, which takes the raw text as input and predicts the abstractive summarization. They also built a discriminator which attempts to distinguish the generated summary from the ground truth summary.

Compressive TS approaches described in Section 2.3.2 improved the informativeness of summaries using sentence and multi-sentence compression approaches to remove irrelevant information and to generate more informative sentences. Therefore, the next section makes a brief overview of sentence compression methods and details the most relevant approaches in MSC.

2.4 Sentence and Multi-Sentence Compression

Sentence Compression (SC) aims at producing a reduced grammatically correct sentence. Compressions may have different CR levels, whereby the lower the Compression Ratio (CR) level, the higher the reduction of the information is. SC can be employed in the contexts of the summarization of documents, the generation of article titles or the simplification of complex sentences, using diverse methods such as tree-based and sentence-based approaches. Tree-based methods compress sentences by making edits to their syntactic trees (Knight and Marcu, 2002; Galley and McKeown, 2007), while sentence-based methods generate compressions directly (McDonald, 2006; Clarke and Lapata, 2007; Filippova et al., 2015; Rush et al., 2015; Miao and Blunsom, 2016).

Knight and Marcu (2002), and Galley and McKeown (2007) parsed the sentences, then generated their compressions by deleting parts of their syntax trees. McDonald (2006), Clarke and Lapata (2007) and Filippova et al. (2015) formulated the SC task by making a binary decision for each word in the source sentences that remain in their compression. Recently, many SC approaches using NN have been developed (Filippova et al., 2015; Rush et al., 2015; Miao and Blunsom, 2016). These methods may generate good results for a single sentence because they combine many complex structures such as RNNs (based on Gated Recurrent Units and Long Short Term Memory), the sequence-to-sequence paradigm and condition mechanisms (e.g., attention mechanism). However, these composite neural networks need huge corpora to learn how to generate compressions (e.g., Rush et al. (2015) used the Gigaword corpus that contains around 9.5 million news) and take a lot of time to accomplish the learning process.

Multi-Sentence Compression (MSC), also coined as Multi-Sentence Fusion, is a variation of SC. Unlike SC, MSC combines the information of a cluster of similar sentences to generate a new sentence, hopefully grammatically correct, which compresses the most relevant data of this cluster. The idea of MSC was introduced by (Barzilay and McKeown, 2005), who developed a multi-document summarizer which represents each sentence as a dependency tree; their approach aligns and combines these trees to fusion sentences. Filippova and Strube (2008) also used dependency trees to align each cluster of related sentences and generated a new tree, this time with ILP, to compress the information. In 2010, Filippova (2010) presented a new model for MSC, simple but effective, which is based on Graph Theory and a list of stopwords. She used a Word Graph (WG) to represent and to compress a cluster of related sentences; the details of this model, which is extended by the work of this thesis, can be found in Section 4.1.1.

Inspired by the good results of the Filippova's method, many studies have used it in a first step to generate a list of the N shortest paths, then have relied on different reranking strategies to analyze the candidates and select the best compression (Boudin and Morin, 2013; Tzouridis et al., 2014; Luong et al., 2015b; Banerjee et al., 2015; Nayeem et al., 2018). Boudin and Morin (2013) developed a reranking method measuring the relevance of a candidate compression using *key phrases*, obtained with the TextRank algorithm (Mihalcea and Tarau, 2004), and the length of the sentence. Another reranking strategy was proposed by (Luong et al., 2015b). Their method ranks the sentences

from the counts of unigrams occurring in every source sentence. ShafieiBavani et al. (2016) also used a WG model; their approach consists of three main components: (i) a merging stage based on Multiword Expressions (MWE), (ii) a mapping strategy based on synonymy between words and (iii) a reranking step to identify the best compression candidates generated using a Part-of-Speech-based language model (POS-LM). Tzouridis et al. (2014) proposed a structured learning-based approach. Instead of applying heuristics as (Filippova, 2010), they adapted the decoding process to the data by parameterizing a shortest path algorithm. They devised a structural support vector machine to learn the shortest path in possibly high dimensional joint feature spaces and proposed a generalized loss-augmented decoding algorithm that is solved exactly by ILP in polynomial time. Nayeem et al. (2018) proposed an extension of the WG to generate compressions. They modelled a cluster of similar sentences as WG and they proposed substitute words for noun and verbs in WG using Paraphrase Database (PPDB 2.0) (Pavlick et al., 2015) and word embeddings. They ranked the compressions using a linear combination of their relevance and abstractiveness. The relevance of sentences is calculated using the TextRank algorithm and the similarity of their sentence embeddings; and the abstractiveness is estimated using a 3-gram language model.

We found two other studies that applied ILP to combine and compress several sentences. Banerjee et al. (2015) developed a multi-document TS system that generated summaries after compressing similar sentences. They used Filippova's method to generate 200 random compressed sentences. Then they created an ILP model to select the most informative and grammatically correct compression. Thadani and McKeown (2013) proposed another ILP model using an inference approach for sentence fusion. Their ILP formulation relies on n-gram factorization and aims at avoiding cycles and disconnected structures.

Another related task is the sentence aggregation that combines a group of sentences, not necessarily with a similar semantic content, to generate a single sentence (e.g., "*The car is here.*" and "*It is blue.*" can be aggregated into "*The blue car is here.*"). This aggregation can be at semantic and syntactic levels (Reape and Mellish, 1999). The aggregation rules can be acquired automatically from a corpus (Barzilay and Lapata, 2006). However, this process is not possible for all situations and the sentence aggregation depends on the sentence planning to combine the sentences.

These works have brought several improvements to TS by generating more informative and readable summaries. However, they restrain to monolingual documents and summaries, and cannot be applied as such for CLTS. Most CLTS works use state-of-the-art MTs to generate translations without integrating these systems into their approaches.

The literature of MT dates back to the early age of computer science and had seen many breakthroughs, among them the development of statistical methods in the 1980s and end-to-end NN models in the 2010s. Therefore, we do not pretend to provide here a thorough insight on the vast literature of MT and we only focus on the most recent representative works before introducing CLTS models in Section 2.6.

2.5 Machine Translation

Nowadays, there are lots of translated text available. Parallel data from European Parliament, books, subtitles of movies and TV series, translated United Nations documents and so on enabled the development of statistical machine translation systems that analyze these parallel datasets to predict the translation of words, sentences and texts.

Statistical Machine Translation combines translation model and language model to translate a text of a source language to a target language. One of the first models to analyze these data was the word-based translation system (IBM models, Figure 2.17(a)) (Brown et al., 1988, 1993; Koehn, 2010). These models consider lexical entries with only one word on either the source-language or target-language side. They use maximum likelihood estimation to estimate a correspondent word in the other language. These models also use fertility, which determines how many foreign words each native word produces, and distortion, which controls how the words are re-ordered, in order to improve the translation quality. However, classical word-based IBM models cannot capture local contextual information and local reordering very well.

Phrase-based translation models operate on lexical entries with more than one word, or n-grams (Figure 2.17(b)) (Och and Weber, 1998; Koehn et al., 2003; Och and Ney, 2004). The option of having multi-word expressions on either the source or target-language side is a significant change from IBM models 1 and 2, which are essentially word-to-word translation models (i.e., they assume that each French word is generated from a single English word). The allowance of n-grams improved the translation quality because of the analysis of multi-word expression enabling the analysis of phrasal verbs, proper names, compound nominals or idioms. The statistical phrase-based translation model generates the English output sentence e_{best} given a foreign input sentence f according to the following equations:

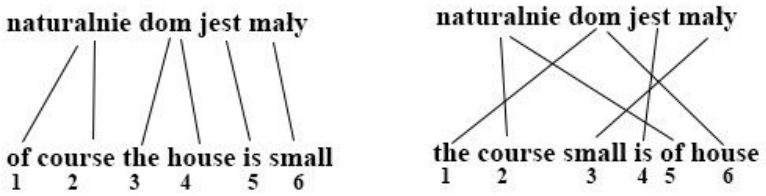
$$e_{\text{best}} = \text{argmax}_e p(e|f) \tag{2.11}$$

$$e_{\text{best}} = \text{argmax}_e p(f|e) p_{\text{LM}}(e) \tag{2.12}$$

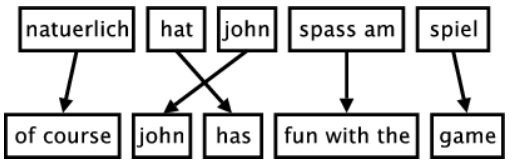
$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(start_i - end_{i-1} - 1) \tag{2.13}$$

where the sentences e and f are segmented into a sequence of I phrases $\bar{e}$ and $\bar{f}$, respectively; $p(f|e)$ is the phrase-based translation model, $p_{\text{LM}}(e)$ is the English language model, ϕ is the phrase translation probability and d is the reordering probability.

Phrased-based models are also limited by how they analyze n-grams inside the context of a sentence; indeed, the meaning of a sentence cannot be determined through an analysis of words or n-grams. Irony and some expressions can change the meaning of a sentence; therefore, a sentence has to be analyzed as a unit to get its real meaning.



(a) Word-based.



(b) Phrase-based

Figure 2.17: *Statistical machine translation models (Source: (Koehn, 2010)).*

The sequence-to-sequence Neural Network model enabled this kind of analysis where a sentence is analyzed entirely to then generate its translation (Figure 2.18). Cho et al. (2014) proposed a sequence-to-sequence model that consists of two recurrent neural networks. One RNN encodes a sequence of symbols into a fixed-length vector representation, and another decodes the representation into another sequence of symbols. The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence.

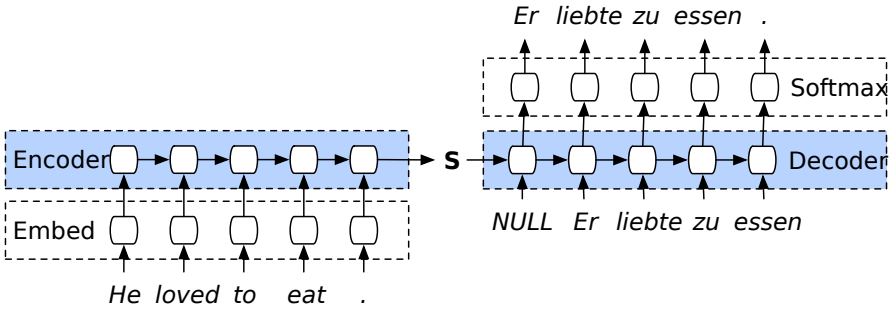


Figure 2.18: *Neural machine translation example using a sequence-to-sequence model (Source:* `https://smerity.com/articles/2016/google_nmt_arch.html`*).*

An attentional mechanism has lately been used to improve Neural Machine Translation (NMT) by selectively focusing on parts of the source sentence during translation. Bahdanau et al. (2014) conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture. They propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. Then, Luong et al. (2015a) proposed two simple and effective classes of attentional mechanism: a global approach which always attends to all source words and a local one that only looks at a subset of source words at a time.

The problem of rare and unknown words is an important issue that can potentially act on the performance of many NLP systems, including traditional count-based and deep learning models (Ling et al., 2015; Gülçehre et al., 2016; Sennrich et al., 2016). Ling et al. (2015) considered the input and output sentences as sequences of characters and Sennrich et al. (2016) as subword units to reduce the out-of-vocabulary words and be able to encode rare and unknown words. Gülçehre et al. (2016) proposed a pointer-generator network to mitigate the problem of rare and unknown words by re-using words from the source sentence or from a vocabulary using a softmax layer.

Finally, Vaswani et al. (2017) proposed a neural network architecture named Transformer which is based solely on attention mechanisms, by dispensing with recurrence and convolutions entirely. This architecture uses stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. This model uses previously generated symbols as additional input when generating the next output. This method achieved the best results for the WMT 2014 English-to-German translation task[3].

Most CLTS systems combine TS and MT approaches to generate cross-lingual summaries. Last improvements in MT also increase the quality of CLTS systems. The next section details the works in CLTS by highlighting how they combine TS and MT approaches to generate cross-lingual summaries.

2.6 Cross-Language Text Summarization

The first studies in cross-language document summarization analyzed the information in only one language (Leuski et al., 2003; Orasan and Chiorean, 2008). Two typical CLTS schemes are the early and the late translations (Figure 2.19). The first scheme first translates the source documents into the target language, then it summarizes the translated documents using only information of the translated sentences. The late translation scheme does the reverse: it first summarizes the documents using abstractive, compressive or extractive methods, then it translates the summary into the target language.

Leuski et al. (2003) proposed an early translation method to generate English headlines for Hindi documents. Orasan and Chiorean (2008) implemented the late translation approach; they produced summaries with the Maximal Marginal Relevance (MMR) method from Romanian news articles and then automatically translated the summaries into English.

Recent methods have improved the quality of cross-language summarization using a translation quality score (Wan et al., 2010; Boudin et al., 2011; Yao et al., 2015b) and the information of the documents in the source and the target languages (Wan, 2011; Zhang et al., 2016). These methods are described in the next two subsections.

[3] http://www.statmt.org/wmt14/

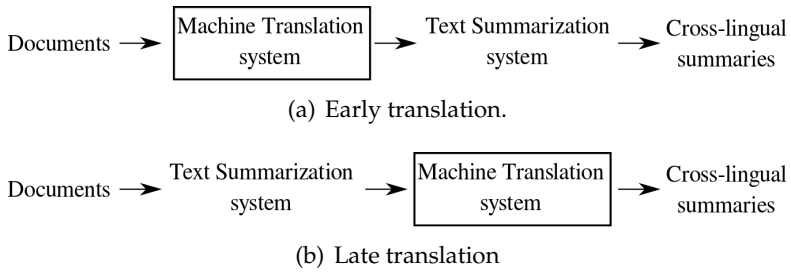


Figure 2.19: *Early and late translations for CLTS.*

2.6.1 Machine Translation Quality

Machine translation evaluation aims to assess the correctness and quality of the translation. Usually, a human reference translation is provided, and various methods and metrics have been developed for comparing the system-translated text and the human reference text.

Another possibility is the use of automatic methods to estimate translation quality (see for example the quality estimation shared task of the WMT conference (Bojar et al., 2017)). The translation quality of a sentence can be estimated at word-level, phrase-level and sentence-level. The estimation at word-level aims to detect errors for each token in MT outputs by deciding if a token is correct in the translation. An incorrect word can cause several errors in the translation, especially in its local context. The estimation at phrase-level is similar to the word-level, i.e. the estimation verifies if a phrase is correct in the translation. Finally, the estimation of translation quality at sentence-level aims to generate scores for the translations according to post-editing effort, i.e. the percentage of needed edits, post-editing time, and so on.

Wan et al. (2010) trained a Support Vector Machine (SVM) regression method to predict the translation quality of a pair of English-Chinese sentences from basic features (such as sentence length, sub-sentence number, percentage of nouns and adjectives) and parse features (such as depth, number of noun phrases and verbal phrases in the parse tree) to generate English-to-Chinese CLTS. They used 1,736 pairs of English-Chinese sentences (English sentences were translated automatically by Google Translate) and computed translation quality scores in a range from 1 to 5 (1 means "very bad" and 5 corresponds to "excellent"). The translation quality and informativeness scores were linearly combined to select the English sentences with both a high translation quality and a high informativeness:

$$score(s_i) = (1 - \lambda) \cdot InfoScore(s_i) + \lambda \cdot TransScore(s_i) \tag{2.14}$$

where $InfoScore(s_i)$ and $TransScore(s_i)$ are the informativeness score and translation quality prediction of the sentence s_i, respectively; and $\lambda \in [0, 1]$ is a parameter controlling the influence of the two factors. Finally, they translated the English summary to form the Chinese summary.

Similarly to Wan et al. (2010), Boudin et al. (2011) used an ϵ-SVR to predict the translation quality score based on the automatic NIST metrics as an indicator of quality. They automatically translated English documents into French using Google Translate, then they analyzed some features (sentence length, number of punctuation marks, perplexities of source and target sentences using different language models, etc.) to estimate the translation quality of a sentence. They incorporated the translation quality score in the PageRank algorithm (Brin and Page, 1998) to calculate the relevance of sentences based on the similarity between the sentences and the translation quality scores to perform English-to-French cross-language summarization (Equations 2.15–2.17).

$$p(v_i) = (1-d) + d \times \sum_{v_j \in pred(v_i)} \frac{score(s_i, s_j)}{\sum_{v_k \in succ(v_i)} score(s_k, s_i)} p(v_i) \tag{2.15}$$

$$score(s_i, s_j) = similarity(s_i, s_j) \times prediction(s_i) \tag{2.16}$$

$$similarity(s_i, s_j) = \frac{\sum_{w \in s_i, s_j} freq(w, s_i) + freq(w, s_j)}{\log(|s_i|) + \log(|s_j|)} \tag{2.17}$$

where d is the damping factor, $prediction(s)$ is the translation quality score of the sentence s, $freq(w, s)$ is the frequency of the word w in the sentence s, $pred(v_i)$ and $succ(v_i)$ are the predecessor and successor vertices of the vertex v_i.

Inspired by the phrase-based translation models, Yao et al. (2015b) proposed a phrase-based model to simultaneously perform sentence scoring, extraction and compression. They designed a scoring scheme for the CLTS task based on a submodular term of compressed sentences and a bounded distortion penalty term to estimate the quality of the translation. Their summary scoring ($F(sum)$) measure was defined over a summary sum as:

$$F(sum) = \sum_{p \in sum} \sum_{i=1}^{count(p, sum)} d^{i-1} g(p) + \sum_{s \in sum} bg(s) + \eta \sum_{s \in sum} dist(pbd(s)) \tag{2.18}$$

where $g(p)$ is the score of phrase p (defined by the frequency of p in the document), $bg(s)$ is the bigram score of sentence s, $pbd(s)$ is the phrase-based derivation of the sentence s and $dist(pbd(s))$ is the distortion penalty term based on the reordering probability of the phrase-based translation models. Finally, d is a constant damping factor to penalize repeated occurrences of the same phrases, $count(p, sum)$ is the number of occurrences of the phrase p in the summary sum and η is the distortion parameter for penalizing the distance between neighboring phrases in the derivation.

2.6.2 Joint Analysis of Source and Target Languages

Wan (2011) proposed to leverage both the information in the source and in the target language for cross-language summarization. In particular, he introduced two graph-

based summarization methods (SimFusion and CoRank) for using both the English-side and Chinese-side information in the task of English-to-Chinese cross-language summarization. The first method linearly fuses the English-side and Chinese-side similarities for measuring Chinese sentence similarity. In a nutshell, this method adapts the PageRank algorithm to calculate the relevance of sentences, where the weight arcs are obtained by the linear combination of the cosine similarity[4] of pairs of sentences for each language:

$$relevance(s_i^{cn}) = \mu \sum_{j \in D, j \neq i} relevance(s_j^{cn}) \cdot \tilde{C}_{ji}^{cn} + \frac{1-\mu}{n} \tag{2.19}$$

$$C_{ij}^{cn} = \lambda \cdot sim_{cosine}(s_i^{cn}, s_j^{cn}) + (1-\lambda) \cdot sim_{cosine}(s_i^{en}, s_j^{en}) \tag{2.20}$$

where s_i^{cn} and s_i^{en} represent the sentence i of a document D in Chinese and in English, respectively, μ is a damping factor, n is the number of sentences in the document and $\lambda \in [0,1]$ is a parameter to control the relative contributions of the two similarity values. C^{cn} is normalized to $\tilde{C}^{cn}$ to make the sum of each row equal to 1.

The CoRank method adopts a co-ranking algorithm to simultaneously rank both English and Chinese sentences by incorporating mutual influences between them (Figure 2.20). It considers a sentence as relevant if this sentence in both languages is heavily linked with other sentences in each language separately (source-source and target-target language similarities) and between languages (source-target language similarity) (Equations 2.21-2.25).

$$\mathbf{u} = \alpha \cdot (\tilde{\mathbf{M}}^{\mathbf{cn}})^T \mathbf{u} + \beta \cdot (\tilde{\mathbf{M}}^{\mathbf{encn}})^T \mathbf{v} \tag{2.21}$$

$$\mathbf{v} = \alpha \cdot (\tilde{\mathbf{M}}^{\mathbf{en}})^T \mathbf{v} + \beta \cdot (\tilde{\mathbf{M}}^{\mathbf{encn}})^T \mathbf{u} \tag{2.22}$$

$$M_{ij}^{\text{en}} = \begin{cases} \text{cosine}(s_i^{en}, s_j^{en}), & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

$$M_{ij}^{\text{cn}} = \begin{cases} \text{cosine}(s_i^{cn}, s_j^{cn}), & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

$$\text{M}_{ij}^{\text{en,cn}} = \sqrt{\text{cosine}(s_i^{cn}, s_j^{cn}) \times \text{cosine}(s_i^{en}, s_j^{en})} \tag{2.25}$$

where $\mathbf{M}^{\mathbf{en}}$ and $\mathbf{M}^{\mathbf{cn}}$ are normalized to $\tilde{\mathbf{M}}^{\mathbf{en}}$ and $\tilde{\mathbf{M}}^{\mathbf{cn}}$, respectively, to make the sum of each row equal to 1. $\mathbf{u}$ and $\mathbf{v}$ denote the saliency scores of the Chinese and English sentences, respectively; α and β specify the relative contributions to the final saliency scores from the information in the same language and the information in the other language, with $\alpha + \beta = 1$.

[4]The cosine similarity between two vectors u and v associated with two sentences is defined by $\frac{u \cdot v}{||u||\ ||v||}$ in the [0,1] range.

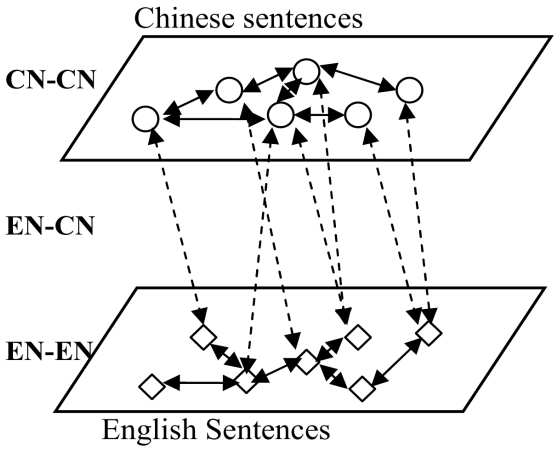


Figure 2.20: *Sentence relationships of CoRank method (Source: (Wan, 2011)).*

Recently, Wan et al. (2018) carried out the cross-language document summarization task by extraction and compression through the ranking of multiple summaries in the target language. They analyzed many candidate summaries in order to produce a high-quality summary for every kind of documents. These candidate summaries were generated using multiple text summarization and machine translation methods, e.g. bilingual submodular function, multiple machine translations and multiple sentence compressions. Their method used a top-K ensemble ranking based on features at several levels and perspectives (word-level, sentence-level, summary-level, readability-related and source-side features) that characterized the quality of a candidate summary.

Wan et al. (2018) who generated extractive and compressive CLTS, Zhang et al. (2016) analyzed Predicate-Argument Structures (PAS) to obtain an abstractive English-to-Chinese CLTS (Figure 2.21). They built a pool of bilingual concepts and facts represented by the bilingual elements of the source-side PAS and their target-side counterparts from the alignment between source texts and Google Translate translations. They used word alignment, lexical translation probability and 3-gram language model to measure the quality and the fluency of the Chinese translation, and the CoRank algorithm (Wan, 2011) to measure the relevance of the facts and concepts in both languages. Finally, summaries were produced by fusing bilingual PAS elements with an Integer Linear Programming (ILP) algorithm to maximize the saliency and the translation quality of the PAS elements. Their ILP model used the pool of bilingual concepts (facts) and their scores to generate summary sentences composed of a concept and at least one core fact.

2.7 Conclusion

NN methods achieved promising results in the last years for text summarization, semantic similarity, sentence compression, machine translation and other NLP applications; however, these methods depend on the training corpora to learn their models.

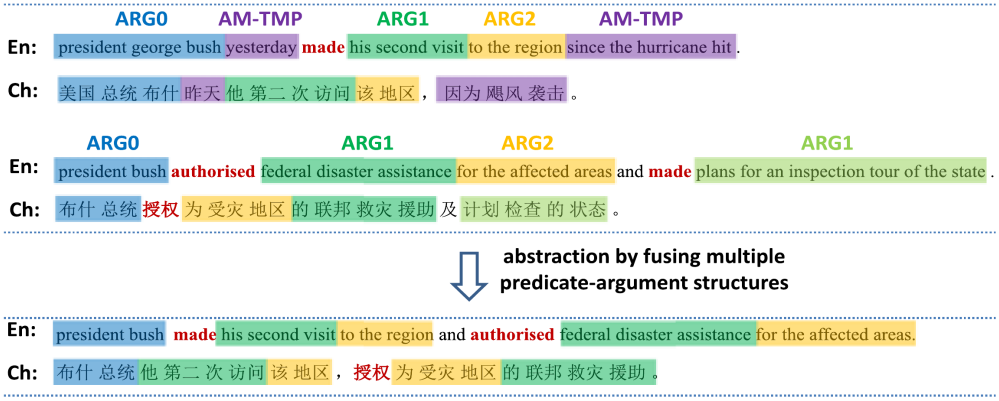


Figure 2.21: *An example of CLTS based on PAS fusing (Source: (Zhang et al., 2016)).*

Unfortunately, most available datasets for these tasks are in English, which reduces the possibility to extend these NN methods for other languages. This thesis aims to generate cross-lingual summaries for several sources and target languages. Therefore, we prefer approaches that are language-independent to easily adapt our framework to other languages, and we only use Neural Networks for the sentence similarity and the sentence compression tasks in the English language. In particular, RNNs and CNNs models were considered for these tasks to build more complex models and to generate better results (Chapter 3).

Among the existing approaches for summarization, abstractive TS methods have a greater capacity to generate summaries more similar to the human abstracts. However, this kind of summarization requests large datasets available in a language to train NN models. On the contrary, extractive summarization approaches do not require specific resources to generate summaries; nevertheless, these extracted sentences may contain redundant and/or non-relevant information, thus reducing the informativeness of summaries. Finally, some compressive methods only require a few resources in several languages to generate summaries. Therefore, we devise a MSC approach optimized to TS which generates compressions guided by keywords and the cohesion of words (Chapter 4). This approach is easily adaptable to other languages and can still improve the informativeness of cross-lingual summaries (Chapter 5).

Chapter 3

Semantic Textual Similarity

Contents

Most works in NLP use a kind of sentence similarity in their methodologies (Mihalcea and Tarau, 2004; Wan, 2011) (Linhares Pontes et al., 2016). For example, IR, TS and CLTS systems use sentence similarity to estimate the relevance of sentences and/or to cluster the sentences by topic.

Semantic Text Similarity (STS) analyzes the degree of semantic similarity between two sentences. It is a difficult issue since languages have numerous ambiguities and synonymous expressions, while sentences may have variable lengths and complex structures. Therefore basic models, e.g. bag-of-words or TF-IDF models, are constrained by their specificities that put aside the role played by the word order and ignore syntactic as well as semantic relationships. Most of them use the cosine similarity measure. Although the cosine similarity using one-hot encoding is one of the most popular sentence similarity measures, it does not analyze the order or the relationship between the words. Recent successes in sentence similarity have been obtained using Neural Networks (CNN (He et al., 2015) and RNN (Kiros et al., 2015; Tai et al., 2015; Mueller and Thyagarajan, 2016)). NN uses a deep analysis of sentences and words to take into account both the semantics and the structure of sentences in order to predict the sentence similarity. Sentence similarity plays a major role in TS (and in this thesis) because it is used in several TS analysis: clustering of sentences, topic identification, sentence relevance, content redundancy and information retrieval.

In this chapter, we describe our system based on NNs to measure the semantic similarity of pairs of sentences (Linhares Pontes et al., 2018d). First, we use a Siamese CNN (more details in Section 3.2) to analyze the local context of words in a sentence and

to generate a representation of the relevance of a word and its neighborhood. Then, we use a Siamese LSTM to analyze the entire sentence based on its words and its local contexts. At last, we predict the semantic similarity of pairs of sentences using the Manhattan distance. We applied our framework on the SemEval dataset for STS assignment and we acquired competitive outcomes demonstrating that our model can give helpful information to enhance the sentence analysis.

This chapter is organized as follows: we make an overview of relevant work for STS in Section 3.1. Next, we detail our approach in Section 3.2. The experimental setup and results are presented in Sections 3.3 and 3.4, respectively. Finally, we give our conclusion and some last remarks in Section 3.5.

3.1 Related Work

Besides being a subjective problem, there are several levels of similitude for two given sentences (one keyword in common, same topic; and same, opposite and independent meanings). STS is analyzed primarily using two approaches: supervised and unsupervised.

Supervised approaches require labeled datasets to learn how to predict the similarity of pairs of sentences. These datasets must be large enough to contain several examples of similarity at the syntactic and lexical dimensions for all sorts of levels of similarities. The syntactic dimension analyzes sentences with different sentence constructions that may share the same meaning. The lexical dimension analyzes the semantics and grammatical roles of words, i.e. the meaning of words can be the same even if words are not the same. Supervised approaches may have good precision in terms of similarity. However, there are few datasets that analyze a few subjects, which limits the generalization of STS.

As regards unsupervised approaches, they do not require labeled datasets. However, they do not have the same precision as supervised methods. Another problem with unsupervised approaches is the prediction of different levels of similarity, which is extremely difficult in the absence of labeled datasets. Similarity measures (e.g. cosine similarity) provide similarity scores in limited range (e.g. $[0, 1]$) to determine if two sentences are similar. However, they do not have a defined measure to predict a similarity level from these scores. Some approaches use pre-trained word embeddings to predict the similarity of sentences. However, word embeddings contain general context of words which can provide similar representations for words with the same context but similar or opposite meanings. For example, Kågebäck et al. (2014) proposed two approaches to represent sentences as sentence embeddings by using pre-trained word embeddings. The first approach represents sentences by calculating the average of their continuous word representations. The second one uses a recursive AE and pre-trained word embeddings to encode the sentences in continuous representations. Then, they calculate the cosine similarity of these sentence embeddings to estimate the relevance of sentence for TS. However, these similarity values do not correlate with the semantic

similarity values.

To deal with the STS task, previous studies have resorted to various features (e.g. word overlap, synonym/antonym), linguistic resources (e.g. WordNet (Miller, 1995) and pre-trained word embeddings) and a wide assortment of learning algorithms (e.g. SVR, regression functions and NN). Among these works, several techniques extract multiple features of sentences and apply regression functions to estimate these similarity scores (Severyn et al., 2013; Lai and Hockenmaier, 2014; Zhao et al., 2014; Bjerva et al., 2014). Lai and Hockenmaier (2014) analyzed distinctive word relations (e.g. synonyms, antonyms, and hyperonyms) with features based on counts of co-occurrences with other words and similarities between captions of images. Zhao et al. (2014) predicted the sentence similarity from syntactic relationship, distinctive content similitudes, length and string features. Severyn et al. (2013) combined relational syntactic structures with SVR. Finally, Bjerva et al. (2014) also utilized a regression algorithm to estimate STS from different features (WordNet, word overlap, and so forth).

The development of NN has improved the results of many NLP applications and especially the STS task (He et al., 2015; Mueller and Thyagarajan, 2016; Tsubaki et al., 2016; Rychalska et al., 2016). Architectures such as RNN and CNN further improve the semantic analysis and the prediction of sentence relatedness. For example, Tsubaki et al. (2016) encode meanings and structures of sentences using word embeddings in a low-dimensional space. Then, they use multiple kernels to learn the semantic similarity of sentences and update the word embeddings.

RNNs differ from other NN models in their ability to process sequential information. They update a memory cell to make sense of data read in a sentence over time. Rychalska et al. (2016) used a RAE and a WordNet grant framework to produce sentence embeddings. They consolidated these embeddings with a SVM classifier to compute a semantic relatedness score. Kiros et al. (2015) proposed a skip-thought model, which feeds each sentence into an RNN encoder-decoder. This model attempts to reconstruct the immediately preceding and following sentences and, subsequently, distinguishes sentences that share semantic and syntactic properties. Finally, they predict STS using a classifier trained on the SICK data based on features derived from differences and products between skip-thought vectors. Socher et al. (2014) used Dependency Tree Recurrent Neural Network (DT-RNN) to embed sentences into a vector space. These NNs focus on the action and on the agents of a sentence to generate sentence embeddings. They also proposed a Semantic Dependency Tree Recurrent Neural Network (SDT-RNN), which embeds a sentence using semantic relations. Long Short Term Memory (LSTM) enhances RNNs to handle long-term dependencies (Mueller and Thyagarajan, 2016; Greff et al., 2015; Tai et al., 2015). The LSTM engineering is made out of a memory cell and non-direct gating units that update its state over time and manage the data stream into/out the cell. Mueller and Thyagarajan (2016) used a Siamese LSTM to encode sentences using pre-trained word embedding vectors. Siamese LSTMs used the same weights to encode sentences and to produce comparable sentence representations for similar sentences. Then, they predicted the closeness of pairs of sentences using the Manhattan distance between the sentence representations. Tai et al. (2015) introduced the Tree-LSTM that is a generalization of LSTM for tree-structured network

topologies. They utilized this Tree-LSTM to encode a couple of sentences and to predict their closeness with a NN that analyzes the distance and the angle between the sentence embeddings.

CNNs have accomplished excellent outcomes in classification (Kim, 2014) and other NLP tasks (Collobert et al., 2011). He et al. (2015) generated sentence embedding using a Siamese CNN architecture with various convolution and pooling operations to extract distinctive granularities of information. Their convolution uses filters that analyze entire word embeddings and each dimension of word embeddings with multiple window sizes. For output of the convolution operation, they applied several pooling types (max, mean, and min). Finally, they predicted the sentence similarity from numerous measurements (horizontal and vertical comparison) to compare local regions of sentence representation.

We join the ideas examined in (Mueller and Thyagarajan, 2016) and (Kim, 2014) to produce more accurate semantic sentence embeddings. The next section presents our model and its characteristics w.r.t. previous work.

3.2 Our Model

A sentence is composed of words which can form phrases and clauses. Examining a sentence and its components helps us to comprehend its meaning. NNs are structures that can inspect relationships between words from multiple points of view. On the one hand, LSTM can recognize and process the semantics of a sentence by investigating the words through time. They update their state to get the gist of the sentence (global context) in the order of words. In this procedure, LSTMs filter unimportant data by retaining just the main information. On the other hand, CNNs use layers with convolution filters that are connected to local features (Kim, 2014). They enable the analysis of a sentence from multiple perspectives (filters). This type of NNs does not have the same concern with the sentence length as LSTMs since CNNs examine all the words of the sentence together. Nonetheless, CNNs do not consider the order of words in their analysis, so these structures cannot investigate sequence relationships in the sentence.

Differently from Mueller and Thyagarajan (2016) that only analyze the general context of words and from He et al. (2015) that do not consider the order of words in the sentences, we analyze the words in two perspectives: general and local contexts. Words are considered through time from the general information of a word (word embedding) and its specific semantic and syntactic features (local context) based on its previous and its following words. We apply a CNN to investigate the local context for each word in a sentence. The CNN analyzes together all the words of the local context and generates their representation as a unique structure. Then, we utilize a LSTM to examine the words of the sentence one by one (Figure 3.1). Our NN has a Siamese structure (He et al., 2015; Mueller and Thyagarajan, 2016), i.e. our CNN^A and our $LSTM^A$ are equal to our CNN^B and our $LSTM^B$, respectively. The following subsection describes our CNN, our LSTM, and our similarity metrics to predict the sentence similarity.

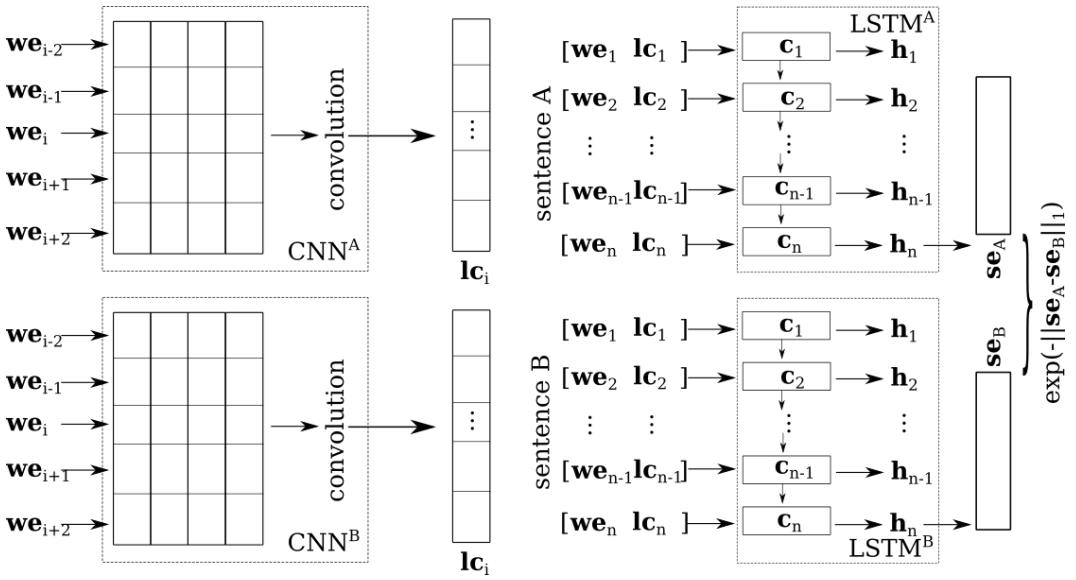


Figure 3.1: *Siamese CNN+LSTM model to calculate the similarity of a pair of sentences.*

Neural Network Architecture

Kim trained a simple CNN on top of pre-trained word vectors for the sentence classification task (Kim, 2014). His simple model composed of one layer of convolution achieved excellent results on multiple benchmarks. Inspired by the good results of CNN in the sentence classification (Kim, 2014), we use a Siamese CNN to generate local contexts for each word in a sentence from its previous and following words. We utilize pre-trained word embeddings[1] to represent these words. Let $\mathbf{we}_i \in \mathbb{R}^k$ be the k-dimensional word vector corresponding to the i-th word in a sentence. A local context of length l (e.g. $l = 5$) is represented as:

$$\mathbf{xl}_i = \mathbf{we}_{i-2} \oplus \mathbf{we}_{i-1} \oplus \mathbf{we}_i \oplus \mathbf{we}_{i+1} \oplus \mathbf{we}_{i+2} \tag{3.1}$$

where $\oplus$ is the concatenation operator. Our convolution operation involves a filter $\mathbf{W} \in R^{lk}$, which is applied to a window of l words to produce a local context ($\mathbf{lc}$). In more details, our CNN generates the local context of word i by:

$$\mathbf{lc}_i = f(\mathbf{W} \cdot \mathbf{xl}_i + \mathbf{b}) \tag{3.2}$$

where $\mathbf{b}$ is a bias term and f is the hyperbolic tangent function. This filter is connected to every sequence of words in a sentence to deliver a local context for all words.

In order to analyze the general and the local contexts of the word i, we concatenate its pre-trained word embeddings $\mathbf{we}_i$ (general semantic and syntactic features that were learned on a large corpus) and its local context $\mathbf{lc}_i$. Our LSTM updates its state $\mathbf{c}_i$ and produces an output $\mathbf{h}_i$ at time step i in a sentence using the equations described

[1]Publicly available at: `https://code.google.com/archive/p/word2vec/`

in (Mueller and Thyagarajan, 2016). The last output of our LSTM $\mathbf{h_n}$ represents the meaning of a sentence.

Diverse similarity metrics (cosine, Euclidean and Manhattan distances) were tested and we acquired the best outcome with the Manhattan distance $\exp(-||\mathbf{se}_A - \mathbf{se}_B||_1) \in [0, 1]$. Since these scores are not optimized for the similarity metric range (1-5), we apply in a post-processing step a regression method using local regression and bandwidth to project our predictions in the correct scale, similarly to (Li and Racine, 2003).

3.3 Experimental Setup

We use the SICK dataset (Marelli et al., 2014) to analyze and to test the performance of our system. This dataset contains 9,840 sentence pairs and we split it in 4,840/2,000/ 3,000 for training/validation/test. Each sentence pair is annotated with a relatedness label $\in$ [1, 5], with 1 indicating that the semantics of the sentences are completely independent and 5 meaning that there is a semantic equivalence, corresponding to the average relatedness judged by 10 different individuals. The gold scores for relatedness are composed of: 923 pairs within the [1,2) range, 1,373 pairs within the [2,3) range, 3,872 pairs within the [3,4) range, and 3,672 pairs within the [4,5] range.

We initialize the CNN and the LSTM weights with small random Gaussian entries. The CNN has filters R^{300} and LSTM has 50-dimensional hidden representations $\boldsymbol{h_t}$ and memory cells $\boldsymbol{c_t}$. We use a forget bias of 2.5 to model long-range dependencies, Adadelta method (Zeiler, 2012) to optimize the parameters, and a learning rate of 0.01. No improvement was measured with deep LSTMs because of the small amount of data. Like (Mueller and Thyagarajan, 2016), we also augmented the training dataset and pretrained our network using the dataset of SemEval 2013 STS task(Agirre et al., 2013).

3.4 Results

In order to understand the relevance of the local context for the sentence similarity, we investigated the original Siamese LSTM without local context and compared it with our method using various lengths for the local context: 3, 5, 7, and 9 (Table 3.1). The original Siamese LSTM analyzes a sentence considering only the general context of words. As expected, the analysis of general and local contexts of words improved the sentence analysis, according to the Pearson's and Pearman's correlation coefficients and the Mean Squared Error (MSE) scores. Short or long local contexts did not generate the best results, which shows that short local context (3 words) did not get enough information about the neighborhood of words and long local context (7 words) includes irrelevant information.

The bottom part of Table 3.1 compares the results of our system and the best state-of-the-art systems. Although our method did not generate the best results, our system is among the top systems and the results were improved with respect to the publicly

Method	r	ρ	**MSE**
Siamese LSTM (Mueller and Thyagarajan, 2016)	*0.8822*	*0.8345*	*0.2286*
Siamese LSTM (publicly available version)[2]	0.8500	0.7860	0.3017
Siamese #local context: 3 + Siamese LSTM	0.8536	0.7909	0.2915
Siamese #local context: 5 + Siamese LSTM	**0.8549**	**0.7933**	**0.2898**
Siamese #local context: 7 + Siamese LSTM	0.8540	0.7922	0.2911
Siamese #local context: 9 + Siamese LSTM	0.8533	0.7890	0.2923
Meaning Factory run1 (Bjerva et al., 2014)	0.8268	0.7722	0.3224
ECNU_run1 (Zhao et al., 2014)	0.8280	0.7689	0.3250
Non-Linear Similarity (Tsubaki et al., 2016)	0.8480	0.7968	0.2904
Constituency Tree LSTM (Tai et al., 2015)	0.8582	0.7966	0.2734
Skip-thought+COCO (Kiros et al., 2015)	0.8655	0.7995	0.2561
Dependency Tree LSTM (Tai et al., 2015)	0.8676	**0.8083**	**0.2532**
ConvNets (He et al., 2015)	**0.8686**	0.8047	0.2606

Table 3.1: *Pearson (r) and Spearman (ρ) correlation coefficients, and Mean Squared Error (MSE) for the test set of STS task.*

available version of the original Siamese LSTM. The post-processing regression step improved MSE by an average of 0.02 for all local context length versions.

In order to illustrate how our local context acts on sentence analysis, Table 3.2 shows at the word level the similarity of a pair of paraphrases: *1-"Her life spanned years of incredible change for women."* and *2-"Mary lived through an era of liberating reform for women."*. For each pair of words taken in both sentences, the similarity measured as a cosine distance[3] is computed either from general word embeddings (Table 3.2.a) or local contexts of length 5 (Table 3.2.b). The first thing to notice is that the two tables have different ranges of values because their dimensional spaces are different; this means that values must be compared inside each table. Analyzing Table 3.2.a shows that word embeddings preserve general semantic and syntactic relationships of words. In this case, the words are more similar to the words that have similar semantics (*1-"Her"*, *2-"Mary"* and *2-"women"*; *1-"life"* and *2-"lived"*; *1-"change"* and *2-"reform"*) and/or have similar syntactic roles (*1-"of"* and *2-"for"*). Table 3.2b highlights that the local context of a word has its semantic and syntactic features based on the words in its window; e.g. the nearest contexts to *1-"life"* are *2-"Mary"*, *2-"lived"*, *2-through* and *2-"women"* since these local contexts have directly (*2-"lived"*) and indirectly (*2-"Mary"*, *2-"through"* and *2-"women"*) similar semantics. This analysis is similar to the syntactic features for the local contexts, e.g. the nearest local context of *1-"for"* are *2-"lived"*, *2-"of"*, *2-"for"* and *2-"woman"*. The relevance of local context is strengthened when we analyze phrasal verbs or multi-word expressions in which meaning depends strongly on their previous and their following words.

[2]We used the public version of Siamese LSTM (Mueller and Thyagarajan, 2016) available at `https://github.com/aditya1503/Siamese-LSTM`, however, we did not get the same results as the ones described in their paper.

[3]The cosine distance between two vectors u and v is defined by $1 - \frac{u \cdot v}{||u||_2||v||_2}$.

Sent. 1 \ Sent. 2	Mary	lived	through	an	era	of	liberating	reform	for	women
Her	0.77	0.93	0.90	0.81	1.04	0.92	0.95	0.91	0.80	0.80
life	0.91	0.70	0.89	0.90	0.82	1.00	0.71	0.86	0.88	0.86
spanned	0.88	0.76	0.81	1.01	0.80	0.85	0.92	1.00	0.89	0.93
years	0.88	0.70	0.94	0.88	0.72	0.86	0.92	0.93	0.81	0.86
of	0.93	0.96	0.96	1.09	0.91	0.00	0.99	1.02	0.82	0.91
incredible	0.94	0.89	0.83	0.94	0.84	0.95	0.74	1.04	0.83	0.97
change	0.97	0.90	0.93	0.92	0.85	0.99	0.80	0.67	0.83	0.92
for	0.96	0.97	0.67	0.79	0.89	0.82	0.88	0.92	0.00	0.89
women	0.81	0.96	0.99	0.93	0.92	0.91	0.79	0.88	0.89	0.00

a. Cosine distance between word embeddings.

Sent. 1 \ Sent. 2	Mary	lived	through	an	era	of	liberating	reform	for	women
Her	0.06	0.08	0.09	0.11	0.16	0.12	0.13	0.13	0.09	0.08
life	0.10	0.08	0.09	0.12	0.11	0.13	0.13	0.14	0.10	0.10
spanned	0.15	0.14	0.11	0.11	0.18	0.14	0.14	0.16	0.13	0.12
years	0.13	0.11	0.08	0.13	0.10	0.12	0.11	0.16	0.09	0.09
of	0.12	0.11	0.10	0.12	0.11	0.09	0.12	0.14	0.13	0.11
incredible	0.12	0.12	0.13	0.14	0.19	0.13	0.03	0.16	0.14	0.09
change	0.14	0.13	0.18	0.15	0.18	0.15	0.16	0.02	0.15	0.13
for	0.10	0.09	0.10	0.11	0.12	0.08	0.11	0.12	0.04	0.08
women	0.09	0.07	0.09	0.11	0.11	0.08	0.09	0.14	0.07	0.01

b. Cosine distance between local contexts of length 5.

Table 3.2: *Cosine distance measured between word embeddings (a.) and between the local contexts of length 5 (b.) for each pair of words of two paraphrases.*

Table 3.3 shows four examples of STS scores for multiple levels of similarities. The first pair of sentences describes an example of active and passive voice, with the same meaning (4.9 golden score). The second case is an example of positive and negative sentences (3.3 golden score). The third example is composed of sentences that do not share the same meaning, having 1.0 golden score. Finally, our method helps to determine the semantic relationship of the phrasal verb "*wipe off*" and the verb "*clean*" in the last example. Our approach improves the Siamese LSTM analysis by generating better scores. The local context helps to better identify not only similar sentences but also the negation and sentences with different meanings. This local information provides LSTM with a smoother analysis of words and how they connect in a sentence.

To sum up, the local context of words refined the general context analysis. Our approach identified more details about the words and their local as well as general contexts, which usually leads to improved STS scores.

3.5 Conclusion

Semantic Textual Similarity is an important task for various NLP applications, e.g. TS, Question-Answering, Information Retrieval, etc. Our system combines CNN and

Pair of sentences	Golden score	Siamese LSTM	Our approach
Fish is being cooked by a woman. *A woman is cooking fish.*	4.9	3.84	**4.05**
The bearded man is not sitting on a train. *The bearded man is sitting on a train.*	3.3	3.49	**3.35**
Someone is playing with a toad. *The trumpet is being played by a man.*	1.0	1.51	**1.46**
I will wash up if you wipe off the table. *I will wash up if you clean the table.*	5.0	3.67	**4.08**

Table 3.3: *Examples of semantic textual similarities using Siamese LSTM and our approach (Siamese #local context: 5 + LSTM).*

LSTM structures to analyze, to identify and to preserve the relevant information in each part of sentences and in the whole sentences. The local context turned out to be useful to get additional information about a word in a sentence and to improve the sentence analysis. In our experiments, the local context improved the prediction of the sentence similarity, by reducing the mean squared error and increasing the correlation scores.

Despite the good results, this approach is limited by the training corpus that is composed of general subjects, simple vocabulary and pairs of sentences that are “easily" comparable. News datasets are composed of long and complex sentences containing a more difficult vocabulary. In this case, the idea of similarity is more subjective and difficult to estimate with scores. We made a few tests with some news sentences to fix a threshold to consider two sentences as similar. Our approach tends to generate better results for short sentences with general subjects, i.e. sentences that have similar structures to the SICK dataset; however, long sentences with complex structures describing specific subjects did not produce good results. In this case, simpler approaches that do not consider the order of words may generate better results. Because of this poor performance with news sentences, we did not use this approach in other applications of this thesis.

Next chapter describes our MSC approach to compress similar sentences in order to reduce the redundancy and to keep the main information of sentences.

Chapter 4

Multi-Sentence Compression

Contents

Summarization systems usually rely on statistical, morphological and syntactic analysis approaches (Torres-Moreno, 2014). Some of them use Multi-Sentence Compression (MSC) in order to produce from a set of similar sentences a small-sized sentence which is both grammatically correct and informative (Filippova, 2010; Banerjee et al., 2015). Although compression is a challenging task, it is appropriate to generate summaries that are more informative than the state-of-the-art extractive methods for TS.

The contributions of this chapter are two-fold. (i) We present a new model for MSC that extends the common approach based on Graph Theory, using vertex-labeled graphs and Integer Linear Programming (ILP) to select the best compression. The vertex-labeled graphs are used to model a cluster of similar sentences with keywords.

(ii) Whereas previous work usually limited the experimental study on one or two datasets, we tested our model on three corpora, each in a different language. Evaluations led with both automatic metrics and human evaluations show that our ILP model consistently generate more informative sentences than two state-of-the-art systems while maintaining their grammaticality. Our approach is able to choose the amount of information to keep in the compression output, through the definition of the maximum compression length.

This chapter is organized as follows: we describe and survey the MSC problem in Section 4.1. Next, we detail our approach in Section 4.2. The experiments and the results are discussed in Sections 4.3 and 4.4. Lastly, we provide the conclusion about MSC and some final comments in Section 4.5.

4.1 Related Work

Following previous studies for MSC (Section 2.4) that rely on Graph Theory with good results, this work presents a new ILP framework that takes into account keywords for MSC. We compare our learning approach to the graph-based sentence compression techniques proposed by Filippova (2010) and Boudin and Morin (2013), considered as state-of-the-art methods for MSC. We intend to apply our method on various languages and independent on linguistic resources or tools specific to languages. This led us to put aside systems which, despite being competitive, rely on resources like WordNet or Multiword expression detectors (ShafieiBavani et al., 2016). Since we borrowed concepts and ideas from Filippova's method, we detail her approach in the next section.

4.1.1 Filippova's method

Filippova (2010) modeled a document D containing n similar sentences $\{s_1, s_2, \ldots, s_n\}$, as a directed Word Graph (WG) $G = (V, A)$. V is the set of vertices (words) and A is the set of arcs (adjacency relationship). Figure 4.1 illustrates the word graph G of the following Portuguese sentences:

1. *George Solitário, a última tartaruga gigante Pinta Island do mundo, faleceu.* (Lonesome George, the world's last Pinta Island giant tortoise, has passed away.)
2. *A tartaruga gigante conhecida como George Solitário morreu domingo no Parque Nacional de Galapagos, Equador.* (The giant tortoise known as Lonesome George died Sunday at the Galapagos National Park in Ecuador.)
3. *Ele tinha apenas cem anos de vida, mas a última tartaruga gigante Pinta conhecida, George Solitário, faleceu.* (He was only about a hundred years old, but the last known giant Pinta tortoise, Lonesome George, has passed away.)
4. *George Solitário, a última tartaruga gigante da sua espécie, morreu.* (Lonesome George, a giant tortoise believed to be the last of his kind, has died.)

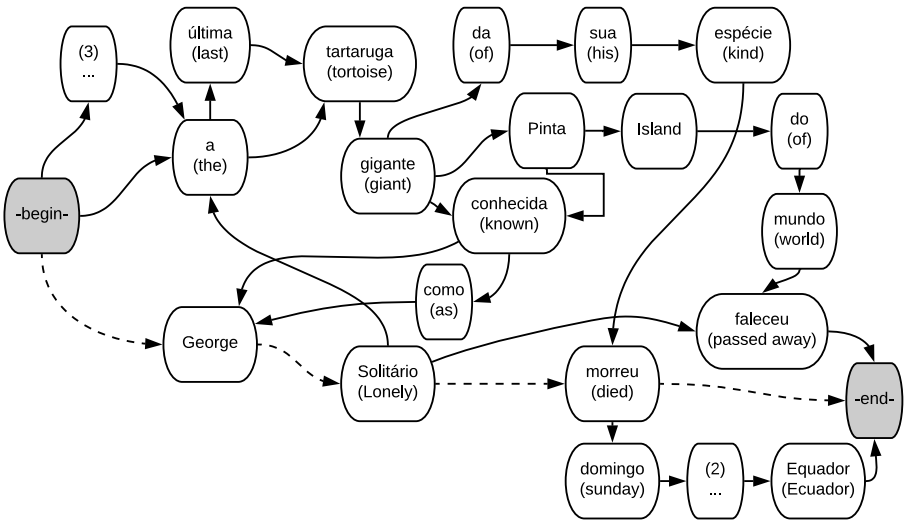


Figure 4.1: *Word graph G generated from the sentences (1) to (4) (without the punctuation and Part-of-Speech (POS) for easy readability). The dotted path represents a possible compression for this WG.*

The initial WG is composed of the first sentence (1) and the vertices –begin– and –end–. For a new sentence, a new vertex is created when a word/POS pair cannot be matched to an existing vertex of G once lowercased. Besides, at most one occurrence of a given word/POS inside a sentence can be associated with a given vertex.

Sentences are individually analyzed and added to G. Each sentence represents a simple path between the –begin– and –end– vertices and its words are inserted in the following order:

1. Non-stopwords for which no candidate exists in the graph or for which an unambiguous mapping is possible;

2. Non-stopwords for which there are several possible candidates in the graph that may occur more than once in the sentence;

3. Stopwords.

In cases 2 and 3, the word mapping is ambiguous because there is more than one vertex in the graph that references the same word/POS. In this case, we analyze the immediate context (the preceding and following words/POSs in the sentence and the neighboring nodes in the graph) or the frequency (i.e., the number of words that were mapped to the considered vertex) to select the best candidate node.

Once vertices have been added, arcs are valued by weights which represent the levels of cohesion between two words in the graph (Equation 4.1). Cohesion is calculated from the frequency and the position of these words in sentences, according to Equation 4.2:

$$w(i,j) = \frac{\text{cohesion}(i,j)}{\text{freq}(i) \times \text{freq}(j)}, \tag{4.1}$$

$$\text{cohesion}(i,j) = \frac{\text{freq}(i) + \text{freq}(j)}{\sum_{s \in D} \text{diff}(s,i,j)^{-1}}, \tag{4.2}$$

where $\text{freq}(i)$ is the word frequency mapped to the vertex i and the function $\text{diff}(s,i,j)$ refers to the distance between the offset positions of words i and j in the sentences s of D containing these two words.

From the graph G, the system calculates the 50 shortest paths that are longer than eight words and have at least one verb. Finally, the system reranks the paths by normalizing the total path weight over their length and selects the path with the lowest score as the best MSC.

4.1.2 Boudin and Morin's method

Boudin and Morin (2013) proposed a method to better evaluate the quality of a sentence and generate more informative compressions from the approach described by Filippova (Section 4.1.1). They used the same Filippova's methodology to generate the 200 shortest paths, which have at least eight words and at least one verb, from the WG. Rather than performing a simple normalization of values as Filippova, they measured the relevance of generated sentences based on key phrases and sentence lengths, according to Equations 4.3 and 4.4:

$$score(p) = \frac{\sum_{i,j \in \text{path}(p)} w(i,j)}{||c|| \times \sum_{kp \in p} score_{kp}(kp)}, \tag{4.3}$$

$$score_{kp}(kp) = \frac{\sum_{w \in kp} \text{TextRank}(w)}{||kp|| + 1}, \tag{4.4}$$

where p is one of the shortest paths calculated by the Filippova's methodology, the $w(i,j)$ is the score between vertices i and j (Equation 4.1), the TextRank algorithm (Mihalcea and Tarau, 2004) calculates the relevance of a word w from its previous and following words, and $score_{kp}(kp)$ is the relevance of the key phrase kp present in the path p. Finally, the sentence with the lowest score is the compression of the cluster of similar sentences.

4.2 Our approach

Filippova's method chooses the path with the lowest score taking into account the level of cohesion between two adjacent words in the document. However, two words with

a strong cohesion do not necessarily have a good informativeness because the cohesion only measures the distance and the frequency of words in the sentences. In this work, we propose a method to concurrently analyze cohesion and keywords in order to generate a more informative and comprehensible compression.

Our method calculates the shortest path from the cohesion of words and grants bonuses to the paths that have different keywords (Linhares Pontes et al., 2016, 2018c). For this purpose, our approach is based on Filippova's method (Section 4.1.1) to model a document D as a graph and to calculate the cohesion of words. In addition, we analyze the keywords of the document to favor hypotheses with meaningful information.

4.2.1 Keyword extraction

Introducing keywords in the graph helps the system to generate more informative compressions because it takes into account the words that are representative of the cluster to calculate the best path in the graph, and not only the cohesion and frequency of words. Keywords can be identified for each cluster with various extraction methods and we study three widely used techniques: LDA (Blei et al., 2003), LSA (Deerwester et al., 1990) and TextRank (Mihalcea and Tarau, 2004). Despite the small number of sentences per cluster, these methods generate good results because clusters are composed of similar sentences with a high level of redundancy. For LDA whose modeling is based on the concept of topics, we consider that the document D describes only one topic since it is composed of semantically close sentences related to a specific news item. A same word or keyword can be represented by one or several nodes in WGs (see Section 4.1.1). In order to prioritize the sentence generation containing multiple keywords and to reduce the redundancy, we add a bonus to the compression score when the compression contains different keywords.

4.2.2 Vertex-Labeled Graph

A vertex-labeled graph is a graph $G = (V, A)$ with a label on the vertices $K = \{0, ..., |K|\}$, where $|K|$ is the number of different labels. This graph type has been employed in several domains such as biology (Zheng et al., 2011) or NLP (Bruckner et al., 2013). In this last study, the correction of Wikipedia inter-language links was modeled as a Colorful Components problem. Given a vertex-colored graph, the Colorful Components problem aims at finding the minimum-size edge sets that are connected and do not have two vertices with the same color.

In the context of MSC, we want to generate a short informative compression where keyword may be represented by several nodes in the word graph. Labels enable us to represent keywords in vertex-labeled graphs and generate a compression without repeated keywords while preserving the informativeness. In this framework, we grant bonuses only once for nodes with the same label to prioritize new information in the compression (Figure 4.2). To make our model coherent, we added a base label (label 0)

for all non-keywords in the word graph. The following section describes our ILP model to select sentences including labeled keywords inside WGs.

4.2.3 ILP Modeling

There are several algorithms with a polynomial complexity to find the shortest path in a graph. However, the restriction on the minimum number $P_{\min}$ of vertices (i.e., the minimum number of words in the compression) makes the problem NP-hard (Garey and Johnson, 1990). Indeed, let v_0 be the –begin– vertex. If $P_{\min}$ equals $|V|$ and if we add an auxiliary arc from –end– vertex to v_0, our problem is similar to the traveling salesman problem, which is NP-hard.

For this work we use the formulation known as Miller-Tucker-Zemlin to solve our problem (Öncan et al., 2009; Thadani and McKeown, 2013). This formulation uses a set of auxiliary variables, one for each vertex in order to prevent a vertex from being visited more than once in the cycle and a set of arc restrictions.

The problem of production of a compression that favors informativeness and grammaticality is expressed as Equation 4.5. In other words, we look for a path (sentence) that has a good cohesion and contains a maximum of labels (keywords).

$$\text{Minimize} \Big(\sum_{(i,j)\in A} w(i,j) \cdot x_{i,j} - c \cdot \sum_{k \in K} b_k \Big) \tag{4.5}$$

where x_{ij} indicates the existence of the arc (i,j) in the solution, $w(i,j)$ is the cohesion of the words i and j (Equation 4.1), K is the set of labels (each representing a keyword), b_k indicates the existence of a word with label (keyword) k in the solution and c is the keyword bonus of the graph.[1]

4.2.4 Structural Constraints

We describe the structural constraints for the problem of consistency in compressions and define the bounds of the variables. First, we consider the problem of consistency which requires an inner and an outer arc active for every word used in the solution, where y_v indicates the existence of the vertex v in the solution.

$$\sum_{i \in \delta^+(v)} x_{vi} = y_v \qquad \forall v \in V, \tag{4.6}$$

$$\sum_{i \in \delta^-(v)} x_{iv} = y_v \qquad \forall v \in V. \tag{4.7}$$

[1]The keyword bonus allows the generation of longer compressions that may be more informative.

The constraints (4.8) and (4.9) control the minimum and the maximum number of vertices ($P_{\min}$ and $P_{\max}$) used in the solution respectively, i.e., the minimum and the maximum number of words in the final compression.

$$\sum_{v \in V} y_v \geq P_{\min}, \tag{4.8}$$

$$\sum_{v \in V} y_v \leq P_{\max}. \tag{4.9}$$

The set of constraints (4.10) matches label variables (keywords) with vertices (words), where $V(k)$ is the set of all vertices with label k.

$$\sum_{v \in V(k)} y_v \geq b_k, \qquad \forall k \in K. \tag{4.10}$$

Equality (4.11) sets the vertex v_0 in the solution.

$$y_0 = 1. \tag{4.11}$$

The restrictions (4.12) and (4.13) are responsible for the elimination of sub-cycles, where u_v ($\forall v \in V$) are auxiliary variables for the elimination of sub-cycles and M is a large number (e.g., $M = |V|$).

$$u_0 = 1, \tag{4.12}$$

$$u_i - u_j + 1 \leq M - M \cdot x_{ij} \qquad \forall (i,j) \in A, j \neq 0. \tag{4.13}$$

Finally, equations (4.14) – (4.16) define the field of variables.

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A, \tag{4.14}$$

$$y_v \in \{0,1\}, \qquad \forall v \in V, \tag{4.15}$$

$$u_v \in \{1,2,\ldots,|V|\}, \qquad \forall v \in V. \tag{4.16}$$

We calculate the 50 best solutions according to the objective (equation 4.5) having at least eight words and at least one verb. Specifically, we find the best solution, then we add a constraint in the model to avoid this solution and repeat this process 50 times to find the other solutions.

The optimized score (Equation 4.5) explicitly takes into account the size of the generated sentence. Contrary to Filippova's method, sentences may have a negative score because we subtract from the cohesion value of the path the introduced scores for keywords. Therefore, we use the exponential function to ensure a score greater than zero. Finally, we select the sentence with the lowest final score (Equation 4.17) as the best compression.

$$\text{score}_{norm}(s) = \frac{e^{\text{score}_{opt}(s)}}{||s||}, \tag{4.17}$$

where $\text{score}_{opt}(s)$ is the score of the sentence s from Equation 4.5.

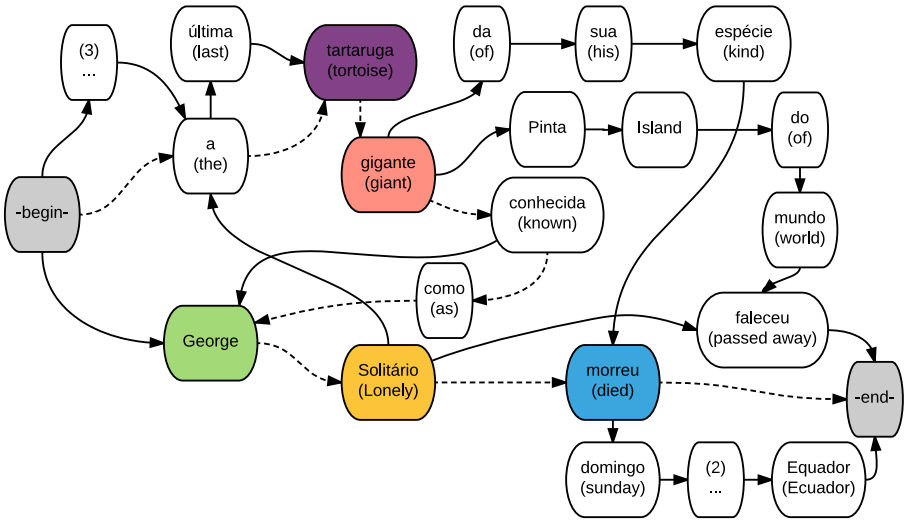


Figure 4.2: *Colored WG generated from the sentences (1) to (4) (without the punctuation and POS for easy readability). The dotted path represents the best compression for this WG and the colored vertices represent the keywords of the document.*

4.3 Experimental Setup

Algorithms were implemented using the Python programming language with the takahe[2] and gensim[3] libraries. The mathematical model was implemented in C++ with the Concert library and we used the solver CPLEX 12.6.[4]

The objective function (see Equation 4.5) involves a keyword bonus. Since each WG can have weight arcs of different values, fixing this bonus is decisive to allow the generation of slightly longer compressions. We tested several metrics (fixed values, the arithmetic average, the median, and the average geometric of the weights arcs of

[2] `http://www.florianboudin.org/publications.html`
[3] `https://radimrehurek.com/gensim/models/ldamodel.html`
[4] `https://www.ibm.com/products/ilog-cplex-optimization-studio`

WG) to define the keyword bonus of the WG and empirically found that the geometric average metric outperformed others.

4.3.1 Evaluation Datasets

Various corpora have been developed for MSC and are composed of clusters of similar sentences from different source news in English, French, Portuguese, Spanish or Vietnamese languages. Whereas the data built by McKeown et al. (2010) and Luong et al. (2015b) have clusters limited to pairs of sentences, the corpora made by Filippova (2010), Boudin and Morin (2013), and Linhares Pontes et al. (2018) contain clusters of at least 7 similar sentences. McKeown et al. (2010) collected 300 English sentence pairs taken from newswire clusters using Amazon's Mechanical Turk, while the corpus introduced in Luong et al. (2015b) is made of 250 Vietnamese sentences divided into 115 groups of similar sentences with 2 sentences by group. McKeown et al. (2010), Luong et al. (2015b), Boudin and Morin (2013), and Linhares Pontes et al. (2018) made their corpora publicly available, but only the data associated with these last two articles are more suited to the multi-document summarization or question-answering tasks because the documents to analyze are usually composed of many similar sentences. Therefore, we use these two corpora in French (Boudin and Morin, 2013), Portuguese and Spanish sentences (Linhares Pontes et al., 2018).

Table 4.1 summarizes the statistics of this set of data having 40 clusters of sentences for each language. The Type-Token Ratio (TTR) indicates the reuse of tokens in a cluster and is defined by the number of unique tokens divided by the number of tokens in each cluster; the lower the TTR, the greater the reuse of tokens in the cluster. The sentence similarity represents the average cosine similarity of the sentences in a cluster.

The French corpus has 3 sentences compressed by native speakers for each cluster, references having a CR of 60%. Like the French corpus, the Portuguese and Spanish corpora are composed of the first sentences of the articles found in Google News[5]. Each cluster is composed of related sentences and was chosen among the first sentence from different articles about Science, Sport, Economy, Health, Business, Technology, Accidents/Catastrophes, General Information and other subjects. A cluster has at least 10 similar sentences by topic and 2 reference compressions made by different native speakers. The average CRs are 54% and 61% for the Portuguese and the Spanish corpora, respectively.

The three languages derive from Latin and are closely related languages. However, they differ in many details of their grammar and lexicon. Moreover, the datasets produced for the three languages are unlike according to several features. First, Linhares Pontes et al. (2018)'s corpus contains a smaller (Portuguese corpus) and a larger (Spanish corpus) dataset in terms of sentences than the French corpus. Besides, the compression ratios of the three datasets indicate that the Portuguese source sentences have more irrelevant tokens. The sentence similarity (Table 4.1, second last line) describes the variability of sentences in the source sentences and in the references, and

[5] `https://news.google.com`

Characteristics \ Datasets	French		Portuguese		Spanish	
	Source	Reference	Source	Reference	Source	Reference
No. tokens	20,224	2,362	17,998	1,425	30,588	3,694
No. vocabulary (tokens)	2,867	636	2,438	533	4,390	881
No. sentences	618	120	544	80	800	160
Avg. sentence length (tokens)	33.0	19.7	33.1	17.8	38.2	23.1
TTR	38.8%	50.1%	33.7%	67.9%	35.2%	43.4%
Sentence similarity	0.46	0.67	0.51	0.59	0.47	0.64
Compression ratio	—	60%	—	54%	—	61%

Table 4.1: *Statistics of the source clusters (source) and their reference compressions (reference) in French, Portuguese and Spanish datasets.*

reflects here that the sentences are slightly more diverse for the French corpus. This translates into a higher TTR observed for the French part (38.8%) than for the two other languages (33.7% and 35.2%).

4.3.2 Automatic and Manual Evaluations

The most important features of MSC are informativeness and grammaticality. Informativeness measures how informational is the generated text. As references are assumed to contain the key information, we calculated informativeness scores counting the n-grams in common between the system output and the reference compressions. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measure developed by Lin (2004) compares the differences between the distribution of words of the candidate summary and a set of reference summaries. The comparison is made splitting into n-grams both the candidate and the reference to calculate their intersection. Standard n-gram values for ROUGE are 1-gram and 2-gram, both expressed as:

$$\text{ROUGE} - n = \frac{\sum_{n-grams} \in \{Sum_{can} \cap Sum_{ref}\}}{\sum_{n-grams} \in Sum_{ref}}, \tag{4.18}$$

where n is the n-gram order, Sum_{can} the candidate summary and Sum_{ref} the reference summary. A third common ROUGE-n variation is ROUGE-SUγ. This ROUGE-2 variation takes into account skip units (SU) $\leq \gamma$. We considered ROUGE-1 and ROUGE-2 to evaluate and compare our system. Like in (Boudin and Morin, 2013), ROUGE metrics are calculated with stopwords removal and French stemming.[6]

Due to limitations of the ROUGE systems that only analyze unigrams and bigrams, we also led a manual evaluation with four native speakers for French, Portuguese and Spanish. The native speakers of each language evaluated the compression in two aspects: informativeness and grammaticality. In the same way as (Filippova, 2010) and (Boudin and Morin, 2013), the native speakers evaluated the grammaticality in a 3-point scale: 2 points for a correct sentence; 1 point if the sentence has minor mistakes;

[6] `http://snowball.tartarus.org/`

0 point if it is none of the above. Like grammaticality, informativeness is evaluated in the same range: 2 points if the compression contains the main information; 1 point if the compression misses some relevant information; 0 point if the compression is not related to the main topic.

4.4 Experimental Assessment

Compression rates are strongly correlated with human judgments of meaning and grammaticality (Napoles et al., 2011). On the one hand, too short compressions may compromise sentence structure, reducing the informativeness and grammaticality. On the other hand, longer compressions may be more interesting for TS when informativeness and grammaticality are decisive features. Consequently, we analyze compression with multiple maximum compression lengths (50%, 60%, 70%, 80%, 90% and ∞, the last value meaning that no constraint is fixed on the output size).

Following the idea proposed by ShafieiBavani et al. (2016) and already implemented with success in other domains such as speech recognition (e.g., (Huet et al., 2010)), we tested the use of a POS-based Language Model (POS-LM) as a post-processing stage in order to improve the grammaticality of compressions. Specifically, for each cluster, the ten best compressions according to our optimized score are reranked by a 7-gram POS-LM trained with the SRILM toolkit[7] on the French, Portuguese and Spanish parts of the Europarl dataset,[8] tagged with TreeTagger (Schmid, 1995).

4.4.1 Results

Since our method strongly depends on the set of keywords to generate informative compressions, we investigate the performance of the three keyword methods (LDA, LSA and TextRank), selecting the 5 or 10 most relevant words. We verified the percentage of keywords generated by these methods that are included in the reference compression (Table 4.2). A significantly higher rate of keywords in the references is observed when using LDA or LSA instead of TextRank. In order to obtain the most relevant words in a cluster with different sizes, we used LDA in our final MSC system to identify 10 keywords for each cluster.

Tables 4.3, 4.4 and 4.5 describe the ROUGE recall scores measured for (Filippova, 2010)'s method (named F10), (Boudin and Morin, 2013)'s method (named BM13) and our method with multiple maximum compression lengths. As for each CR setup the size of the outputs to evaluate is comparable, the recall scores are preferred in this case to measure the information retained in compressions. First, let us note that CRs effectively observed may differ from the fixed value of P_{max}. For example, a 50% threshold leads to real CRs of 38% to 40% for all languages, while a 80% level creates new

[7] http://www.speech.sri.com/projects/srilm/

[8] http://www.statmt.org/europarl/

Methods \ Corpora	French	Portuguese	Spanish
LDA: 5 keywords	**91%**	**88%**	**85%**
LSA: 5 keywords	90%	87%	81%
TextRank: 5 keywords	69%	55%	58%
LDA: 10 keywords	**84%**	**70%**	**76%**
LSA: 10 keywords	**84%**	69%	73%
TextRank: 10 keywords	56%	44%	50%

Table 4.2: *Percentage of keywords included in the reference compression.*

sentences with real CRs between 53% and 60%. Interestingly, our system obtained better ROUGE recall scores than both baselines in all languages for comparable compression lengths. If we prioritize meaning, our method with no explicit constraint on the maximum compression length (ILP:∞) improved the compression quality with a small increase of the compression length (compression ratio between 55.4% and 65.9%). Instead, we can limit the length and generate compressions that are shorter and have still better ROUGE scores than the baselines.

Methods	ROUGE-1	ROUGE-2	CR
F10	0.5971	0.4072	51.3%
BM13	0.6740	0.4695	59.8%
ILP:50%	0.4763	0.3039	39.1%
ILP:60%	0.5990	0.4101	47.4%
ILP:70%	0.6420	0.4206	53.5%
ILP:80%	0.6783	0.4573	60.0%
ILP:90%	0.6981	**0.4758**	61.8%
ILP:∞	**0.7010**	0.4751	62.6%

Table 4.3: *ROUGE recall scores for multiple maximum compression lengths using the French corpus.*

Based on these results, a further analysis was done for the 80% and ∞ configurations. Table 4.6[9] describes the results for the French, Portuguese and Spanish corpora using ROUGE F-scores. The first two columns display the evaluation of the two baseline systems; the ROUGE scores measured with our method using either 80% or ∞ maximum compression lengths are shown in the next two columns and the last two columns respectively. The outputs produced by all these systems for two sample clusters in Spanish and Portuguese can be found in Section 4.4.3.

Globally, all versions of our ILP method outperform both baselines according to ROUGE F-scores for the Portuguese and Spanish corpora, and our ILP systems (ILP:80%

[9] Although we used the same system and data as Boudin and Morin (2013) for the French corpus, we were not able to exactly reproduce their results. The ROUGE F-scores given in their article are close to ours for their system: 0.6568 (ROUGE-1), 0.4414 (ROUGE-2) and 0.4344 (ROUGE-SU4), but using F10 we measured higher scores than them: 0.5744 (ROUGE-1), 0.3921 (ROUGE-2) and 0.3700 (ROUGE-SU4).

Methods	ROUGE-1	ROUGE-2	CR
F10	0.5354	0.2935	52.2%
BM13	0.6304	0.3493	69.1%
ILP:50%	0.4689	0.2521	40.0%
ILP:60%	0.5369	0.2967	48.1%
ILP:70%	0.5652	0.3088	54.0%
ILP:80%	0.6056	0.3321	59.0%
ILP:90%	0.6341	0.3492	64.6%
ILP:∞	**0.6407**	**0.3546**	65.9%

Table 4.4: *ROUGE recall scores for multiple maximum compression lengths using the Portuguese corpus.*

Methods	ROUGE-1	ROUGE-2	CR
F10	0.4437	0.2631	43.2%
BM13	0.5167	0.2981	61.2%
ILP:50%	0.3814	0.1990	38.7%
ILP:60%	0.4594	0.2651	45.3%
ILP:70%	0.5050	0.2922	50.2%
ILP:80%	0.5191	0.2982	53.2%
ILP:90%	0.5242	0.2982	54.4%
ILP:∞	**0.5305**	**0.3036**	55.4%

Table 4.5: *ROUGE recall scores for multiple maximum compression lengths using the Spanish corpus.*

and ILP:∞) obtained similar results to BM13 for the French corpus. The POS-LM post-processing improved the ROUGE scores for Portuguese and Spanish, however, it reduced the ROUGE scores of our methods for the French corpus. Table 4.7 displays the average length, the compression ratio and the average number of keywords that are kept in the final compression. F10 generated the shortest compressions for all corpora, our approach producing outputs of an intermediate length with respect to BM13, except for the French corpus for which ILP:∞ generated a bit longer compressions. As expected, the POS-LM post-processing, which favors grammaticality, tended to select sentences with fewer keywords.

We also led a manual evaluation to study the informativeness and grammaticality of compressions. We measured the inter-rater agreement on the judgments we collected, obtaining values of Fleiss' kappa of 0.423, 0.289 and 0.344 for French, Portuguese and Spanish respectively. These results show that human evaluation is rather subjective. Questioning evaluators on how they proceed to rate sentences reveals that they often made their choice by comparing outputs for a given cluster.

Table 4.8 shows the manual analysis that ratifies the good results of our system. Informativeness scores are consistently improved by the ILP method, whereas grammaticality results measured on the three systems are similar. Besides, statistical tests

Metrics	F10	BM13	ILP:80%	ILP:80%+LM	ILP:∞	ILP:∞+LM
French						
ROUGE-1	0.6384	0.6674	0.6630	0.6418	**0.6730**	0.6460
ROUGE-2	0.4423	**0.4672**	0.4487	0.4187	0.4567	0.4179
ROUGE-SU4	0.4297	**0.4602**	0.4410	0.4152	0.4511	0.4136
Portuguese						
ROUGE-1	0.5388	0.5532	0.5668	0.5763	0.5700	**0.5811**
ROUGE-2	0.2971	0.3029	0.3105	0.3112	0.3132	**0.3249**
ROUGE-SU4	0.2938	0.2868	0.3060	0.3149	0.3057	**0.3210**
Spanish						
ROUGE-1	0.5004	0.5140	0.5422	**0.5500**	0.5425	0.5442
ROUGE-2	0.2983	0.2960	0.3128	**0.3195**	0.3109	0.3194
ROUGE-SU4	0.2847	0.2801	0.2973	**0.3052**	0.2963	0.3047

Table 4.6: *ROUGE F-scores for MSC using the French, Portuguese and Spanish corpora. The best ROUGE results are in bold.*

show that this enhancement regarding informativeness and grammaticality is significant for Spanish corpus. For the Portuguese and Spanish corpora, our method obtained the best results for informativeness and grammaticality with shorter compressions. For the French corpus, F10 obtained the highest value for grammatical quality, while BM13 generated more informative compressions. Finally, the reranking method proposed by BM13 based on the analysis of *key phrases* of candidate compression improves informativeness, but not to the same degree as our ILP model. This more moderate enhancement can be related to the fact that this reranking method is limited to candidate sentences generated by F10.

4.4.2 Discussion

Short compressed sentences are appropriate to summarize documents; however, they may remove key information and prejudice the informativeness of the compression. For instance, for the sentences that would be associated with a higher relevant score by the TS system, producing longer sentences would be more appropriate. Generating longer sentences makes easier to keep informativeness but often increases difficulties to have a good grammatical quality while combining different parts of sentences. Depending on the kind of cluster short compressions can be generated or not with good informativeness scores. In that respect, the system has to adapt its analysis to generate long or short sentences.

F10 produced the shortest compressions for all corpora but its outputs have the worst informativeness score. BM13 improves these results; however, their compressions are longer than F10 (for all corpora) and our system (for the Portuguese and the Spanish corpora). For Spanish, the informativeness scores of all versions of our method are statistically better than F10, and the version ILP:∞+LM is statistically better than

Metrics	F10	BM13	ILP:80%	ILP:80%+LM	ILP:∞	ILP:∞+LM
French						
Avg. Length	16.9 ± 5.1	19.7 ± 6.9	19.8 ± 4.8	19.5 ± 4.9	20.6 ± 5.5	20.8 ± 5.8
Comp. Ratio. (%)	51.3	59.8	59.9	59.2	62.6	63.1
Keywords	6.8	7.7	8.3	7.9	8.5	8.1
Portuguese						
Avg. Length	17.3 ± 5.3	22.9 ± 6.3	19.5 ± 4.0	19.4 ± 4.4	21.8 ± 5.5	20.5 ± 5.0
Comp. Ratio. (%)	52.2	69.1	59.0	58.7	65.9	62.2
Keywords	7.0	8.5	8.2	8.0	8.9	8.3
Spanish						
Avg. Length	16.5 ± 6.4	23.4 ± 8.4	20.3 ± 5.9	20.9 ± 5.2	21.1 ± 7.0	23.4 ± 7.3
Comp. Ratio. (%)	43.2	61.2	53.2	54.7	55.4	61.2
Keywords	5.8	6.9	7.7	7.6	7.9	7.9

Table 4.7: *Compression length (#words), standard deviation and number of used keywords computed on the French, Portuguese and Spanish corpora.*

both baselines for this corpus. Given the small difference of informativeness between BM13 and our ILP approach for the French and the Portuguese corpora, we analyzed how informativeness and CR are related to define which method obtains the best results. For the French corpus, it is complicated to bring forward a best system because the second baseline, ILP:80% and ILP:∞ have similar informativeness scores for close CRs. For Portuguese, BM13 and all versions of our system achieve similar informativeness scores for the Portuguese corpus, whereas our method generates significantly shorter compressions with an absolute decrease in the range 3.0–10.1 points. All in all, our approach generates compressions that are better than the two baselines for the Portuguese and the Spanish corpora, and shares a quality similar to BM13 for the French corpus.

Reviewing results of Tables 4.7 and 4.8, the informativeness scores and keywords appear as related, i.e., the higher the number of keywords the higher the informativeness score. Let us note that according to how clusters are made (with respect to the size and the amount of information), they can have a number of effective keywords that are more or less lower than the fixed number of 10. The number of keywords and informativeness scores are related except for BM13 on the French corpus that used fewer keywords than our method and generated more informative compressions.

The POS-LM post-processing does not improve significantly the compression quality of our method. This post-processing maintain or enhance grammaticality for all corpora, except for the ILP:∞+LM for Portuguese corpus, and informativeness for the Portuguese and the Spanish corpora. The biggest difference between these two versions of all methods is on the Spanish corpus (differences of 0.1 and 0.14 are observed for informativeness and grammaticality, respectively), for which the POS-LM version generated a longer version (CR is increased by 5.8 points), which is related to the improvement of informativeness.

Metrics	F10	BM13	ILP:80%	ILP:80%+LM	ILP:∞	ILP:∞+LM
French						
Informativeness						
Score 0	20%	10%	14%	16%	14%	14%
Score 1	36%	31%	32%	35%	27%	34%
Score 2	44%	59%	54%	49%	59%	52%
Avg.	1.25 ± 0.8	$\mathbf{1.48 \pm 0.7}$	1.40 ± 0.7	1.33 ± 0.7	1.45 ± 0.7	1.39 ± 0.7
Grammaticality						
Score 0	6%	7%	12%	8%	10%	10%
Score 1	23%	29%	36%	29%	35%	36%
Score 2	71%	64%	52%	63%	55%	54%
Avg.	$\mathbf{1.65 \pm 0.6}$	1.56 ± 0.6	1.44 ± 0.7	1.55 ± 0.6	1.45 ± 0.7	1.44 ± 0.7
Portuguese						
Informativeness						
Score 0	9%	7%	8%	5%	7%	8%
Score 1	30%	16%	18%	22%	12%	13%
Score 2	61%	77%	74%	73%	81%	79%
Avg.	1.51 ± 0.7	1.70 ± 0.6	1.66 ± 0.6	1.68 ± 0.6	$\mathbf{1.74 \pm 0.6}$	1.71 ± 0.6
Grammaticality						
Score 0	9%	8%	6%	5%	4%	7%
Score 1	21%	18%	18%	21%	15%	17%
Score 2	70%	74%	76%	74%	81%	76%
Avg.	1.61 ± 0.6	1.66 ± 0.6	1.71 ± 0.6	1.69 ± 0.6	$\mathbf{1.76 \pm 0.5}$	1.68 ± 0.6
Spanish						
Informativeness						
Score 0	24%	26%	12%	11%	10%	10%
Score 1	49%	31%	39%	36%	39%	29%
Score 2	27%	43%	49%	53%	51%	61%
Avg.	1.02 ± 0.7	1.16 ± 0.8	1.36 ± 0.7 **	1.41 ± 0.7 **	1.40 ± 0.7 **	$\mathbf{1.50 \pm 0.7}$ **††
Grammaticality						
Score 0	11%	18%	12%	8%	10%	6%
Score 1	26%	33%	35%	36%	35%	29%
Score 2	63%	49%	53%	56%	55%	65%
Avg.	1.51 ± 0.7	1.30 ± 0.8	1.40 ± 0.7	1.48 ± 0.6	1.45 ± 0.7	$\mathbf{1.59 \pm 0.6}$ †

Table 4.8: *Manual evaluation of compression (ratings are expressed on a scale of 0 to 2). The best results are in bold (* and ** indicate significance at the 0.01 and the 0.001 level using ANOVA's test related to F10, respectively; † and †† indicate significance at the 0.01 and the 0.001 level using ANOVA's test related to BM13, respectively).*

4.4.3 Multi-Sentence Compression Example

We analyzed two cluster examples in Spanish and Portuguese to illustrate the differences between the tested methods.

Spanish Cluster

The Spanish cluster (Table 4.9) is composed of 20 similar sentences. The vocabulary of this cluster is composed of 880 tokens and this cluster has a TTR of 33.3%. F10 generated the shortest compression; however, the sentence has missing information. The second baseline system and our method without post-processing generated incorrect compressions. Our method without post-processing generated a sentence with relevant keywords but it is not correct. The post-processing selected a more grammatical compression without reducing informativeness. The top 10 keywords selected by LDA were : *vuelo*, *cuba*, *fort*, *lauderdale*, *unidos*, *primer*, *jetblue*, *comercial*, *clara* and *florida*.

Portuguese Cluster

Table 4.10 displays a cluster composed of 11 Portuguese sentences with a TTR of 37% and a vocabulary of 351 tokens. In this case, F10 did not generate the shortest compression and has incorrect information. The second baseline, which post-processes the outputs of the first one, was not able to correct the errors. Almost all versions of our method generated the shortest and the most informative compressions related to the text. Our method without post-processing generated the best compression. The post-processing selected a more grammatically correct sentence, while its information is incorrect. The top 10 keywords selected by LDA were : *tesla*, *solarcity*, *milhões*, *2,6*, *solar*, *empresa*, *carros*, *fabricante*, *dólares* and *motors*.

4.5 Conclusion

Multi-Sentence Compression aims to generate a short informative text summary from several sentences with related and redundant information. Previous works built word graphs weighted by cohesion scores from the input sentences, then selected the best path to select words of the output sentence. We introduced in this study a model for MSC with two novel features. Firstly, we extended the work done by Boudin and Morin (2013) that introduced keywords to post-process lists of N-best compressions. We proposed to represent keywords as labels directly on the vertices of word graphs to ensure the use of different keywords in the selected paths. Secondly, we devised an ILP modeling to take into account these new features with the cohesion scores, while selecting the best sentence. The compression ratio can be modulated with this modeling, by selecting for example a higher number of keywords for the sentences considered essential for a summary.

Source document	
El vuelo 387 de la aerolínea estadounidense JetBlue inauguró una nueva era en el transporte entre ambos países, al partir desde Fort Lauderdale (Florida, sureste) cerca de las 10:00 locales (14H00 GMT), y llegar a Santa Clara, 280 Km al este de La Habana, a las 10:57. Un avión de pasajeros de la línea aérea JetBlue despegó este miércoles a Cuba desde el aeropuerto Internacional de Fort Lauderdale en lo que viene a ser el primer vuelo regular entre Estados Unidos y la isla caribeña desde 1961, en un nuevo hito en la nueva fase de relaciones entre Washington y La Habana. La aerolínea JetBlue inaugurará los vuelos directos comerciales el 31 de agosto con un viaje entre Fort Lauderdale, Florida, hasta el aeropuerto de Santa Clara, a unos 270 kilómetros al este de La Habana, reportó la compañía estadunidense.	
Reference	
La aerolínea JetBlue Airways Corp inauguró el 31 de agosto los vuelos directos entre Estados Unidos y Cuba tras 50 años de suspensión . (*The airline JetBlue Airways Corp opened on August 31 direct flights between the United States and Cuba after 50 years of suspension.*)	
Compressions	
F10:	la aerolínea jetblue inauguró este miércoles a cuba el primer vuelo inaugural .
BM13:	el aeropuerto de fort lauderdale , florida , sureste de estados unidos y cuba desde 1961 partió este miércoles el primer vuelo inaugural .
ILP:80%	el aeropuerto de fort lauderdale , florida , sureste de estados unidos y cuba desde 1961 partió este miércoles el primer vuelo inaugural .
ILP:80%+LM	la aerolínea jetblue inauguró este miércoles el primer vuelo desde fort lauderdale , florida , sureste de estados unidos a cuba desde 1961 .
ILP:∞	el aeropuerto de fort lauderdale , florida , sureste de estados unidos y cuba desde 1961 partió este miércoles el primer vuelo inaugural .
ILP:∞+LM	la aerolínea jetblue inauguró este miércoles el primer vuelo desde fort lauderdale , florida , sureste de estados unidos a cuba desde 1961 .

Table 4.9: *MSC example in Spanish showing the first 3 sentences among 20 source sentences and 1 of 3 available references.*

Our methodology was evaluated on three corpora built from Google news: a first one in French which had been built and used in (Boudin and Morin, 2013), a second and a third one in Portuguese and in Spanish (Linhares Pontes et al., 2018). Automatic measures with the ROUGE package were supplemented with a manual evaluation carried out by human judges in terms of informativeness and grammaticality. We showed that keywords are important features to produce valuable compressed sentences. The paths selected with theses features generate results consistently improved in terms of informativeness while keeping up their grammaticality.

Source document	
A Tesla fez uma oferta de compra à empresa de serviços de energia solar SolarCity por mais de 2300 milhões de euros. A Tesla Motors, fabricante de carros elétricos, anunciou aquisição da SolarCity por US$ 2,6 bilhões. A fabricante de carros elétricos e baterias Tesla Motors disse nesta segunda-feira (1) que chegou a um acordo com a SolarCity para comprar a instaladora de painéis solares por US$ 2,6 bilhões, em um grande passo do bilionário Elon Musk para oferecer aos consumidores um negócio totalmente especializado em energia limpa, informou a Reuters.	
Reference	
A Tesla Motors anunciou acordo para comprar a SolarCity por US$ 2,6 bilhões. (*Tesla Motors has announced an agreement to buy SolarCity for US$ 2.6 billion.*)	
Compressions	
F10	a solarcity para comprar a instaladora de painéis solares por us$ 2,6 bilhões .
BM13	a solarcity para comprar a instaladora de painéis solares por us$ 2,6 mil milhões de euros .
ILP:80%	a tesla vai comprar a solar solarcity por 2,6 mil milhões de euros .
ILP:80%+LM	a solarcity para comprar a instaladora de painéis solares por 2,6 mil milhões de euros .
ILP:∞	a tesla vai comprar a solar solarcity por 2,6 mil milhões de euros .
ILP:∞+LM	a solarcity para comprar a instaladora de painéis solares por 2,6 mil milhões de euros .

Table 4.10: *MSC example in Portuguese showing the first 3 sentences among 11 source sentences and 1 of 2 available references.*

Chapter 5

Cross-Language Text Summarization

Contents

Cross-Language Text Summarization (CLTS) aims to generate a summary of a document where the summary language differs from the document language. The huge amount of information available on the Internet made it easier to be up to date on the news in the world. However, some information and viewpoints exist in languages that are unknown by readers. CLTS enables people who are not fluent in the source language to comprehend these data in a simple way.

The methods developed for CLTS can be classified, like the Text Summarization (TS) domain, depending on whether they are extractive, compressive or abstractive (Torres-Moreno, 2014). The extractive TS selects complete sentences that are supposed to be the most relevant of the documents; the compressive TS generates a summary by compression of sentences through the removal of non-relevant words; lastly, the abstractive TS generates a summary with new sentences that are not necessarily contained in the original texts.

Many of the state-of-the-art methods for CLTS are of the extractive class. They mainly differ on how they compute sentence similarities and alleviate the risk that translation errors are introduced in the produced summary. Among these models,

the CoRank method, which is characterized by its ability to simultaneously incorporate similarities between the original and translated sentences, turns out to be effective (Wan, 2011). These extractive approaches generate cross-lingual summaries with irrelevant information, which reduces their informativeness. Recent compressive and abstractive CLTS approaches have improved the informativeness of these summaries. (Yao et al., 2015b; Zhang et al., 2016; Wan et al., 2018). However, these approaches use specific linguistic resources that limit them to a pair of languages, making it difficult to adapt these approaches to other languages.

We present a new modular framework to generate compressive cross-lingual summaries for several languages. Our framework combines sentence and multi-sentence compression methods to compress and improve the informativeness of sentences and, consequently, summaries.

Next subsections describe our two approaches to generate cross-lingual summaries. We first analyze French-to-English cross-lingual generation using two sentence compression methods with Multi-Word Expression (MWE) (Section 5.1). Then, we generalize this approach to be able to generate cross-lingual summaries for several pairs of languages (Section 5.2).

5.1 Compressive French-to-English Cross-Language Text Summarization

Inspired by the compressive TS methods in monolingual analysis (Qian and Liu, 2013; Li et al., 2013, 2014; Yao et al., 2015a; Filippova et al., 2015; Banerjee et al., 2015; Niu et al., 2017), we adapt sentence and multi-sentence compression methods for the French-to-English CLTS problem to just keep the main information. Long Short Term Memory (LSTM) model is built to analyze a sentence and decide which words remain in the compression. We also use an Integer Linear Programming (ILP) formulation to compress similar sentences while analyzing both grammaticality and informativeness.

The remainder of this section is organized as follows. Section 5.1.1 presents our compressive CLTS approach. Section 5.1.2 reports the results achieved on the MultiLing 2011 dataset for the French-to-English task and shows that our method, particularly with the use of ILP for multi-sentence compression, outperforms the state of the art according to the ROUGE metrics. Finally, conclusions are set out in Section 5.1.3.

5.1.1 Our Proposition

Following the CoRank-based approach proposed by (Wan, 2011), we use his joint analysis of documents in both languages (source and target languages) to select the most relevant sentences. We expanded this method in three ways.

Firstly, we take into account Multi-Word Expression (MWE) when computing similarities between sentences. These MWEs are very common in all languages and pose

significant problems for every kind of NLP (Moirón and Tiedemann, 2006). Their use in the context of CLTS helps the system to comprehend the semantic content of sentences. To realize a chunk-level tokenization, we used the Stanford CoreNLP tool for the English side (Manning et al., 2014). This annotator tool, which integrates jMWE[1] (Kulkarni and Finlayson, 2011), detects various expressions, e.g., phrasal verbs ("*take off*"), proper names ("*San Francisco*"), compound nominals ("*cultivated plant*") or idioms ("*rain cats and dogs*"). Unfortunately, the tools developed for languages other than English have a lower coverage for MWEs.

A second evolution of the CoRank-based approach is the use of a Multi-Sentence Compression (MSC) method to generate more informative compressed outputs from similar sentences. For this purpose, the sentences are grouped in clusters based on their similarity in both languages. For each cluster with more than one sentence, which is common in the case of multi-document summarization, an MSC method guided by keywords is applied to build a sentence with the core information of the cluster (Linhares Pontes et al., 2016, 2018c).

A third extension of the approach relies on compression techniques of a single sentence by deletion of words (Filippova et al., 2015). Still with the idea to generate more informative summaries, sentence compression is applied for sentences that stand alone during the clustering step required by the MSC step.

The following subsections describe in detail the architecture of our system.

Preprocessing

Initially, French texts are translated into English using the Google Translate system, which is at the cutting edge of the statistical translation technology and was used in the majority of the state-of-the-art CLTS methods. Then, chunks are identified inside the English texts with the Stanford CoreNLP.

Finally, sentences are clustered according to their similarities, sentences with a similarity score bigger than the threshold θ remaining in the same group. The similarity score of a pair of sentences i and j is defined by the cosine similarity in both languages:

$$sim(i,j) = \sqrt{\text{cosine}(s_i^{fr}, s_j^{fr}) \times \text{cosine}(s_i^{en}, s_j^{en})} \tag{5.1}$$

where s_i^{fr} and s_i^{en} represent a sentence i in the French and English languages.

Sentence and Multi-Sentence Compression

To avoid the accumulation of errors that would appear in a translation-compression-translation pipeline, we restrict the sentence and multi-sentence compressions to the

[1] `http://projects.csail.mit.edu/jmwe/`

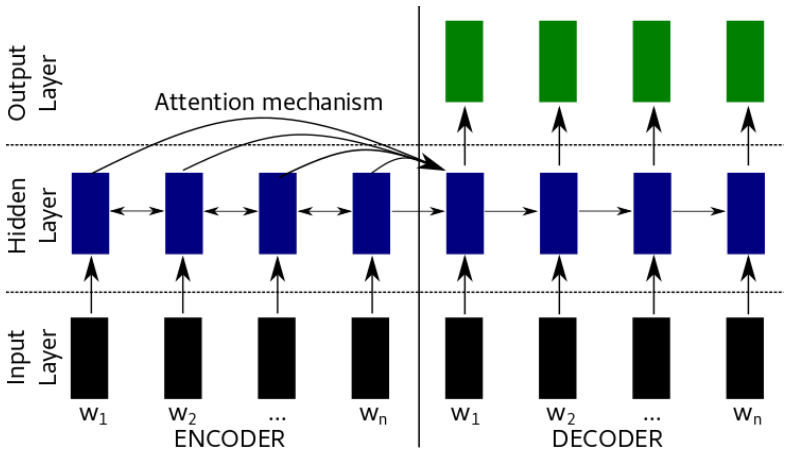


Figure 5.1: *The words are represented by the word embedding representations in the input layer. The attention mechanism improves the decode processing. The output layer is composed of 0 (remove), 1 (remain) or <pad>.*

sentences in the target language.

Sentence Compression

Sentence Compression (SC) problem is here seen as the task to delete non-relevant words in a sentence (Yao et al., 2015a; Filippova et al., 2015; Tran et al., 2016). In a similar way to (Tran et al., 2016), our method extends the LSTM model described in (Filippova et al., 2015) to compress sentence by deletion of words. In few words, our model follows a sequence-to-sequence paradigm using the attention mechanism to verify which words of a sentence c remain in the compression (Figure 5.1). The words in a sentence c are represented by their word embeddings. Then, a first LSTM encodes this sentence (Hochreiter and Schmidhuber, 1997) and a second LSTM with attention mechanism generates the sequence of the words that are kept in the compression. The attention mechanism decides which input region to focus on in order to generate the next output (Bahdanau et al., 2014).

LSTM with attention mechanism is composed of input i_t, control state c_t and memory state m_t that are updated at time step t (Equations 5.2-5.11).

$$x_t = [we_t, c_t] \tag{5.2}$$

$$i_t = \text{sigm}(W_1 x_t + W_2 h_{t-1}) \tag{5.3}$$

$$i'_t = \tanh(W_3 x_t + W_4 h_{t-1}) \tag{5.4}$$

$$f_t = \text{sigm}(W_5 x_t + W_6 h_{t-1}) \tag{5.5}$$

$$o_t = \text{sigm}(W_7 x_t + W_8 h_{t-1}) \tag{5.6}$$

$$m_t = m_{t-1} \odot f_t + i_t \odot i'_t \tag{5.7}$$

$$h_t = m_t \odot o_t \tag{5.8}$$

where the operator $\odot$ denotes element-wise multiplication, we_t is the word embedding of the word at the time step t, c_t is the context vector, the matrices $W_1, ..., W_8$ and the vector h_0 are the parameters of the model, and all the non-linearities are computed element-wise. The context vector at time t c_t is calculated as a sum of all hidden states of the encoder weight:

$$c_t = \sum_{j=1}^{T} \alpha_{tj} \cdot h_E^j \tag{5.9}$$

$$r_{tj} = v_\alpha^T \tanh(W_\alpha h_{t-1} + U_\alpha h_E^j) \tag{5.10}$$

$$\alpha_{tj} = softmax(r_{tj}) \tag{5.11}$$

where the probability α_{tj} represents the importance of each hidden state of the encoder h_E^j in the prediction of the current state h_t. Contrary to (Filippova et al., 2015), we analyze the sentence at the chunk level, so we remove a chunk only if all words of this chunk were deleted in the SC process described above.

Multi-Sentence Compression

For the clusters that have more than a sentence, we use a Chunk Graph (CG) to represent them and an ILP method to compress these sentences in a single, short, and hopefully correct and informative sentence. Among several state-of-the-art MSC methods (Filippova, 2010; Banerjee et al., 2015; Niu et al., 2017)(Linhares Pontes et al., 2016, 2018c), our system incorporates our previous ILP model for the MSC (Chapter 4) to generate compressions of clusters of similar sentences, but instead of restricting to single words (Word Graph) we also consider multi-word chunks (Chunk Graph). We also use LDA to identify the keywords at the global (all texts of a topic) and local (cluster of similar sentences) levels to have the gist of a document and of a cluster of similar sentences.

CoRank Method

The CoRank method adopts a co-ranking algorithm to simultaneously rank both French and English sentences by incorporating mutual influences between them. We use the CoRank method (Section 2.6) to calculate the relevance of sentences. In order to avoid the accumulation of errors that would be generated by a translation-compression- translation pipeline, the similarity is computed from the uncompressed versions of sentences and that is only in the last summary generation step that compressed sentences are used.

Finally, as usual for TS, a summary is generated with the most relevant sentences and the sentences redundant with the ones that have already been selected are put aside.

5.1.2 Experimental Results

In order to analyze the performance of our method, we compare it with the early translation, the late translation, the SimFusion and the CoRank methods (Wan, 2011). Following the idea presented by Wan (2011), the early and late translations are based on the SimFusion method, the differences between the systems being on the similarity metrics (Equation 2.20) computed either in the target language (early translation) or in the source language (late translation) (Wan, 2011). We analyzed the SimFusion method with λ = 0.75[2]. The CoRank method uses $\alpha = \beta = 0.5$. We generated three versions of our approach, named Compressive CLTS (CCLTS): SC, MSC and SC+MSC. The first version uses the SC method to compress sentences, the MSC method compresses clusters of similar sentences and extracts the rest of the sentences, and the last version applies both MSC to clusters of similar sentences and SC to other sentences.

We only compress sentences with more than 15 words and we preserve compressions with more than 10 words to avoid short outputs with little information. The MSC method selects the 10 most relevant keywords per topic and the 3 most relevant keywords per cluster of similar sentences to guide the compression generation. All systems generate summaries composed of 250 words with the most relevant sentences, while the redundant sentences are discarded. We apply the cosine similarity measure with a threshold θ of 0.5 to create clusters of similar sentences for MSC and to remove redundant sentences in the summary generation.

We use the pre-trained word embeddings[3] with 300-dimensional embeddings and an LSTM model with only one layer with 256-dimensional embeddings. Our Neural Network is trained on the publicly released set of 200,000 sentence-compression pairs[4].

[2]SimFusion achieved better results with $\lambda = 0.75$ for Chinese-to-English CLTS in (Wan, 2011).

[3]Publicly available at: `code.google.com/p/word2vec`

[4]`https://github.com/google-research-datasets/sentence-compression/tree/master/data`

Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
baseline.early	0.4165	0.1021	0.1607
baseline.late	0.4142	0.1023	0.1589
SimFusion	0.4173	0.1035	0.1606
CoRank	0.4623*	0.1321	0.1926*
CCLTS.SC	0.4352	0.1259	0.1809
CCLTS.MSC	**0.4743***	**0.1369**	**0.1947***
CCLTS.SC+MSC	0.4517	0.1311	0.1852

Table 5.1: *ROUGE F-scores for the French-to-English CLTS using the MultiLing Pilot 2011 dataset. * indicates the results are statistically better than baselines and the SimFusion method with a 0.05 level.*

Dataset

We used the MultiLing Pilot 2011 dataset (Giannakopoulos et al., 2011) derived from publicly available WikiNews English texts. This dataset is composed of 10 topics, each topic having 10 source texts and 3 reference summaries. Each reference summary contains a maximum of 250 words. Native speakers translated this dataset into Arabic, Czech, French, Greek, Hebrew and Hindi languages. Specifically, we use English and French texts to test our system.

Automatic Evaluation

As references are assumed to contain the key information, we calculated informativeness scores counting the *n*-grams in common between the compression and the reference compressions using the ROUGE system (Section 4.3.2). In particular, we used the F-score metrics ROUGE-1, ROUGE-2 and ROUGE-SU4.

Table 5.1 shows the ROUGE F-scores achieved by each system using the MultiLing Pilot 2011 dataset. The baselines, especially the late translation approach, have the worst scores. Similarly to the results described in (Wan, 2011), CoRank outperforms SimFusion. The analysis of the output of the CCLTS versions brought to light that the SC version removed relevant information of sentences, achieving lower ROUGE scores than CoRank. CCLTS.MSC generated more informative summaries and leads to the best ROUGE scores. Finally, the SC+MSC version obtains better results than other systems but still does not reach the highest ROUGE scores measured when using MSC alone.

Manual Evaluation

Considering the limitations of the automatic evaluation to analyze the grammaticality and the informativeness of cross-lingual summaries, three annotators manually eval-

Methods	Informativeness		Grammaticality	
	Average	Std. Dev.	Average	Std. Dev.
baseline.early	2.9	0.8	3.9	0.5
baseline.late	2.8	0.7	4.0	0.5
SimFusion	2.9	0.7	4.0	0.5
CoRank	3.3	0.4	**4.3**	**0.7**
CCLTS.SC	3.2	0.6	3.7	0.7
CCLTS.MSC	**3.5**	**0.4**	4.1	0.7
CCLTS.SC+MSC	3.1	0.7	3.4	1.0

Table 5.2: *Manual evaluation scores for the French-to-English CLTS using the MultiLing Pilot 2011 dataset.*

uated the cross-lingual summaries in two aspects: grammaticality and informativeness. The informativeness is rated with scores from 1 (summaries without relevant information) to 5 (summaries with the main information of source documents). The grammaticality also has the same range; summaries with several errors has score 1 and summaries without grammatical errors has score 5.

Table 5.2 shows the manual evaluation of cross-lingual summaries. All versions of our system ratified the good results of automatic evaluation and generated more informative summaries than the early, the late and the SimFusion. Our system using MSC obtained the highest score for informativeness. As regards the grammaticality, CoRank generated more grammatical compressions but our MSC approach achieved scores similar to other extractive baselines, which proves the generation of compressive summaries with a good grammaticality. Our SC method removed relevant information, which reduced the informativeness and the grammaticality of compressions. Section 5.1.2 provides the analysis of informativeness and grammaticality of an example using the CoRank and all versions of our approach.

Example Analysis

We carried out the analysis of an example of French-to-English CLTS extracted from the Multilingual Pilot dataset. Table 5.3 shows a reference summary of the cluster of source documents that describes the capture of fifteen sailors and marines by Iranian border guards. Tables 5.4-5.7 show the cross-lingual summaries generated by the CoRank and the three versions of our system. The extractive cross-lingual summary generated by the CoRank method is presented in Table 5.4. Even using an extractive approach, this summary contains some grammatical mistakes.

Our SC method compresses all sentences, generating short compressions (Table 5.5). SC method attempts to reproduce the principle of its training dataset that eliminates the words of the first sentences of news to reproduce their title. This procedure normally works well when the source sentence has a direct sentence with straight ideas.

Two years after the seizure of Royal Navy personnel by Iran, two inquiries, that examined the British Ministry's of Defence handling, identified weaknesses in training, communications and the handling of intelligence as well as "collective failure of judgement". The fifteen sailors and marines, from the frigate HMS Cornwall, were captured by Iranian border guards on March 23 in the Persian Gulf, while they were inspecting, in accordance with UN Security Council Resolution 1723, a ship believed to be smuggling cars into Iraq. The UK insisted they were operating in Iraqi waters, while Iran claimed they entered illegally into Iran's territorial waters and that they could face charges of espionage. If those charges were brought against them, the result would be heavy punishment by current Iranian law. On 28 March, British Prime Minister froze all bilateral business deals with Iran. The next day, Iran announced that it will "suspend" the releasing of 15 British personnel, due to the political ballyhoo by London. The EU called the Iranian seizure a "clear breach" of international law. Meanwhile, footage of all 15 British personnel had been broadcast on Iranian TV, with one of the sailors saying that the soldiers were in Iranian waters at the time of their detainment. The British government claimed that the confessions were extracted under duress. Few days later, Iranian President announced that he would free them as a "gift to the British people". The fifteen British navy personnel landed at Heathrow on 5 April, after thirteen days of captivity.

Table 5.3: *Reference summary.*

On Thursday, British Prime Minister Tony Blair said in a television interview that if the 15 sailors and soldiers who were arrested by the Iranian forces were not released, then Britain would be forced to "enter a new phase". operations, and that Iran only has a few days to free the 15 soldiers and sailors. According to Reuters, the United Kingdom sent a 15-page preliminary statement to the UN Security Council "deploring" the continued arrest and support for the British position that soldiers were operating in Iraqi waters as members of the United Nations Security Council. the Iraqi Multinational Force under the mandate of the Security Council ... and at the request of the Iraqi government. The Iranian National Security Council has announced it will "suspend" the release of 15 British sailors and soldiers arrested by Iranian forces on March 23. The defense minister said Royal Navy sailors were "engaged in routine boarding operations in Iraqi territorial waters" and completed their inspection of the suspect ship when they were surrounded by Iranian forces. He added that he had "asked Mr. Blair not to prosecute these 15 soldiers because they confessed their penetration of Iranian territorial waters," apparently implying that the British military were on a secret mission in Iranian waters, and should not have confessed to the television being in there.

Table 5.4: *Cross-lingual summary generated by the CoRank method.*

However, the source sentences have complex syntactic structures and different ways to explain the facts, which produces summaries with grammatical errors but also with less relevant information. CCLTS.SC generated shorter summaries because we remove short sentences (fewer than 10 words) and redundant sentences from the summaries.

The British government has asked for the release of the military. the forces were in Iranian waters , and continues . First-hand information on the capture and detention by Iran of the 15 Royal crew British similar to the two that were seized by Iran on March 23, 2007. Errors identified in the response to Iran's capture of Royal Navy soldiers. Iranian media said the British sailors had "shouted for joy" at the news. The Iranian government initially located the incident in Iraqi waters. Britain says they will not negotiate the release of their soldiers. "HMS Cornwall frigates and soldiers were inspecting, in accordance with UN Security Council Resolution 1723 The president said "[after the meeting] they [were] free. All 27 members of the union agreed on the content of the communiqué. British sailors detained by Iran will be "tried for espionage" Sunday, March 25, 2007. We want to resolve in peace and dialogue the disagreements we have with your government.

Table 5.5: *Cross-lingual summary generated by the CCLTS.SC method.*

Table 5.6 shows the cross-lingual summary for CCLTS.MSC. This summary is composed of three compressions and other sentences are extracted from the source documents. These compressions enabled the generation of summaries with more subjects than CoRank. Unfortunately, the clusters of similar sentences have sizes and levels of similarity between the sentences smaller than the MSC dataset (Chapter 4). These characteristics may generate summaries with sentences that combine different subjects which can reduce their concision and readability.

A video of the 15 sailors and soldiers aired on the iranian forces were in iranian waters when they were arrested. The United Kingdom has frozen all bilateral economic relations with Iran until the 15 british sailors and soldiers arrested by Iranian forces on march 23. The defense minister said Royal Navy sailors were "engaged in routine boarding operations in Iraqi territorial waters" and completed their inspection of the suspect ship when they were surrounded by Iranian forces. He added that he had "asked Mr. Blair not to prosecute these 15 soldiers because they confessed their penetration of Iranian territorial waters," apparently implying that the British military were on a secret mission in Iranian waters, and should not have confessed to the television being in there. *Iranian president Mahmoud Ahmadinejad announced that the 15 British sailors as a gift to the British people.* The United Kingdom is ready to move to "a new phase" if British soldiers and sailors are not released by Iran in the days that follow. Two investigations into the capture of Royal Navy soldiers by Iran in March 2007 determined that it was not the result of "a point of failure or human error of a particular individual, but rather than an unfortunate accumulation of factors "and that it resulted in a" collective error of judgment "by allowing those who were involved to be paid to detail these events in front of the media.

Table 5.6: *Cross-lingual summary generated by the CCLTS.MSC method.*

Finally, Table 5.7 describes the cross-lingual summary of SC+MSC. The compressions generated by MSC improved the informativeness of SC version; however, this

summary contains the combination of errors generated by MSC and SC which reduced the grammatical quality of summaries.

A video of the 15 sailors and soldiers aired on the Iranian forces were in Iranian waters when they were arrested. The United Kingdom has frozen all bilateral economic relations with Iran until the 15 British sailors and soldiers arrested by Iranian forces on march 23. Iranian president Mahmoud Ahmadinejad announced that the 15 British sailors as a gift to the British people. Cornwall frigate were inspecting, in accordance with UN security council resolution 1723. The Australian reported that a website "operated by associates of Mahmoud Ahmadinejad" declared that the 15 British soldiers who had been arrested by the Iranian revolutionary guards could be accused of espionage. The British government has asked for the release of the military. Iran said Tuesday that soldiers and sailors are being treated "humanly" and that they are "in good health". Errors identified in the response to Iran's capture of royal crew. what needs to be done when engaging with people like the Iranian government understand that sanctions can be taken if they are not prepared to be reasonable. The UE reiterates its call for the immediate and unconditional release of British royal navy soldiers. British similar to the two that were seized by Iran on March 23, 2007. united, a boat into Iraq which proved unfounded after inspection when Iranian ships surrounded the sailors.

Table 5.7: *Cross-lingual summary generated by the CCLTS.SC+MSC method.*

Discussion

The lower results of the early and late translations with respect to other systems prove that the texts in each language provide complementary information. It also establishes that the analysis of sentences in the target language plays a more important role to generate informative cross-lingual summaries. As seen for English-to-Chinese CLTS (Wan, 2011), the CoRank method generates better results than the baselines and SimFusion because it considers the information in each language separately and together, while the baselines restrict the analysis of sentence similarity to one language separately and the SimFusion method analyzes only the cross-language sentence similarity.

It is expected that a piece of information found in several texts is relevant for a topic. In accordance with this principle, the MSC method looks for repeated information and generates a short compression with selected keywords that summarize the main information. The two kinds of keywords (global and local) guide MSC to generate compression linked to the main topic of the documents and to the specific information presented in the cluster. With respect to CoRank, our MSC version improved the informativeness of summaries by generating shorter sentences with the main information.

With regard to SC, this compression method eliminated much relevant information in our experiments. This observation may be explained by the reduced size of the corpus we used to train our NN (200,000 parallel sentence-compression instance), while the system described in (Filippova et al., 2015) could benefit from a corpus of about two million instances. In this case, the SC approach attempts to compress all kinds of sentences (simple and complex grammatical); however, the small training dataset do not

have enough compression examples of long sentences with several subjects and complex syntactic structure. Besides, this training dataset is not suitable for compressing all kinds of sentences because its parallel sentence-compression instances are composed of first sentences of news and their titles. These instances are simpler than compressing other news sentences that have more complex structures. A possible solution to overcome these problems is the use of tree-based and sentence-based SC approaches, such as (Galley and McKeown, 2007; Clarke and Lapata, 2007), to generate more correct and informative compressions for complex sentences.

Whereas the CCLTS.MSC version leaves unchanged the sentences that do not have similar sentences, the SC+MSC version involves the SC model to compress these sentences. As the CCLTS.SC system has lower performance than the pure extractive CoRank method, the SC+MSC also had lower results than MSC version.

A difference between the SC and MSC approaches is that MSC uses global and local keywords to guide the compression by preserving the main information, while the SC method does not realize this kind of analysis. The SC method compresses first sentences of news that normally describe the main idea of the news in a straight way. However, we applied SC for all kinds of sentences, e.g. sentences with complex syntactic structure and/or with several subjects. Our approach generates poor results for these kinds of sentences. Another difference between them is that MSC does not need a training corpus to generate compressions.

To sum up, the joint analysis of both languages with CoRank helps the generation of cross-lingual summaries. On the one hand, the SC model deletes relevant information, thereby reducing the informativeness of summaries. On the other hand, the MSC method proves to be a good alternative to compress redundant information and to preserve relevant one. Finally, the CCLTS.MSC greatly improves the ROUGE scores and significantly outperforms the baselines and the SimFusion methods.

5.1.3 Conclusion

The proposed system analyzes a document in both languages to extract all the relevant information. Then, it applies two kinds of methods to compress sentences. Unlike the sentence compression system (CCLTS.SC) that needs a large training dataset to generate compressions of good quality, the multi-sentence compression version of our system (CCLTS.MSC) generates better ROUGE results than extractive Cross-Language Text Summarization systems.

Next section describes the adaptation of this approach to generate cross-lingual summaries for several pairs of languages.

5.2 A Multilingual Study of Compressive Cross-Language Text Summarization

Previous works analyzed the CLTS only between two languages for a given dataset, which does not demonstrate the stability of methods for different texts and languages. We adapt our approach to perform CLTS for several languages. More precisely, we modified the creation of chunks and we simplified our MSC method to be able to analyze several languages and compress small clusters of similar sentences. To demonstrate the stability of our system, we extend the MultiLing Pilot dataset (Giannakopoulos et al., 2011) with two Romance languages (Portuguese and Spanish) to test our system to generate {French, Portuguese, Spanish}-to-{English, French} cross-lingual summaries. Finally, we carried out an automatic evaluation to make a systematic performance analysis of systems, which details the characteristics of each language and their impacts on the cross-lingual summaries (Linhares Pontes et al., 2018b).

We present the adaptation of our approach in Section 5.2.1. Then, we describe our extension of the MultiLing dataset for the Spanish and Portuguese languages in Section 5.2.2. Finally, we analyze the performance of our new system for French, Portuguese and Spanish (Section 5.2.3); and we set up our conclusions in Section 5.2.4.

5.2.1 New Approach

In order to simplify and to extend the analysis for several languages, we only use two Multi-Word Expressions for the English target language. We use syntactic patterns to create chunks: $< (ADJ)^*(NP|NC)+ >$ for English and $< (ADJ) * (NP|NC) + (ADJ)* >$ for French, where ADJ stands for adjective, NP for proper noun and NC for common noun. Unfortunately, we have not found any available dataset for sentence compression in other languages; therefore, we restrict the use of compressive methods to MSC. Finally, compresses versions of sentences were considered in the CoRank method instead of the only original versions, in order to estimate the relevance of sentences for summary generation.

5.2.2 Datasets

In order to extend the analysis of Romance languages of MultiLingual dataset, English source texts were translated into the Portuguese and Spanish languages by native speakers[5]. Specifically, we use English, French, Portuguese, and Spanish texts to test our system.

[5]The extension of the MultiLing Pilot 2011 dataset is available at: `http://dev.termwatch.es/~fresa/CORPUS/TS/`

5.2.3 Evaluation

Table 5.8 describes ROUGE F-scores obtained by each system to generate French summaries from English, Portuguese, and Spanish source texts. Despite using the information from both languages, the SimFusion method achieved comparable results with respect to the early and late approaches. On the contrary, CoRank and our approach consistently obtained better results (difference of 0.005 in ROUGE-1) than other baselines (difference of 0.02 in ROUGE-1) for all languages. MSC improved CoRank by generating more informative compressions for all languages. The last two lines show that chunks did not significantly improve ROUGE scores (similar ROUGE scores).

Methods	**English**			**Portuguese**			**Spanish**		
	R-1	**R-2**	**R-SU4**	**R-1**	**R-2**	**R-SU4**	**R-1**	**R-2**	**R-SU4**
baseline.late	0.4190	0.0965	0.1588	0.4403	0.1128	0.1746	0.4371	0.1133	0.1738
baseline.early	0.4223	0.1007	0.1631	0.4386	0.1110	0.1743	0.4363	0.1143	0.1729
SimFusion	0.4240	0.1004	0.1637	0.4368	0.1105	0.1735	0.4350	0.1125	0.1723
CoRank	0.4733	0.1379	0.1963	0.4723	0.1460	0.2006	0.4713	0.1387	0.1942
Our approach	**0.4831**	0.1460	**0.2030**	**0.4784**	0.1511	**0.2045**	**0.4825**	0.1481	0.2050
Our approach w/o chunks	0.4817	**0.1463**	0.2021	**0.4784**	**0.1518**	0.2044	0.4805	**0.1486**	**0.2056**

Table 5.8: *ROUGE F-scores (R-1= ROUGE-1, R-2: ROUGE-2, R-SU4: ROUGE-SU4) for cross-lingual summaries from English, Portuguese, and Spanish languages to French language.*

The Multiling dataset is composed of 10 topics in several languages; however, these topics are expressed in different ways for each language. These dissimilarities imply a variety of vocabulary sizes and sentence lengths, and, consequently, of outputs of the MT system from each source language (Table 5.9). The biggest difference in the statistics is between English source texts and its French translation vocabulary. French translations significantly increased the vocabulary from English source texts and the number of words. These translations also are longer than source texts, except for the Spanish that has similar characteristics. The difference in morphology between the English and French languages explains the difference in the vocabulary sizes of the texts. For example, the English language has less verb tenses and conjugations than the French language. Our simple syntactic pattern created similar numbers of chunks for all languages with the same average length. However, the addition of these simple chunks did not significantly improve the informativeness of our compressions because of the small size of clusters which reduces the possibility to generate different kinds of compressions.

These differences also act on the clustering process and the MSC method. Table 5.10 details the number and the average size of clusters with at least two French sentences translated from each source language. French translations from Portuguese produced the shortest compressions (18.6 words) while compressions from Spanish had the highest compression ratio. The size and the length of clusters are correlated to the compression ratio. Large clusters with long sentences have more probability to generate shorter

Characteristics	English		Portuguese		Spanish	
	Source	Fr-Transl.	Source	Fr-Transl.	Source	Fr-Transl.
No. words	36109	39960	37339	39302	40440	40269
No. vocabulary (tokens)	8077	8770	8694	8572	8808	8744
No. sentences	1816	1816	2002	2002	1787	1787
Sentence length	19.9	22.0	18.6	19.6	22.6	22.5
No. chunks	–	1615	–	1579	–	1606
Average length of chunks	–	2.1	–	2.1	–	2.1

Table 5.9: *Statistics of datasets and their translation to French.*

compressions. With respect to other languages, the similarity of the sentences translated from English is lower, which leads to fewer clusters. Summaries from Spanish have a larger proportion of compressions in the summaries than other languages. The higher the amount of clusters, the higher the number of compressions present in the summaries.

Characteristics	English	Portuguese	Spanish
No. clusters	50	70	75
Average size of clusters	2.2	2.7	2.8
Average length of clustered sentences	29.1	25.6	35.4
Average length of compressions	21.7	18.6	23.5
Average number of compressions in summaries	0.7	0.9	1.3
Average compression ratio of compressions	74.6%	72.6%	66.4%

Table 5.10: *Statistics about clusters and compressions for French translated texts.*

We apply a similar analysis for the generation of English summaries from French, Portuguese, and Spanish source texts. As observed before for French summaries, the joint analysis still outperformed other baselines (Table 5.11). While CoRank obtained a large range of ROUGE scores among different languages (ROUGE-1 between 0.4602 and 0.4715), our approach obtained the best ROUGE scores for all languages with a small difference of ROUGE scores (ROUGE-1 between 0.4743 and 0.4725), which proves that our method generates more stable cross-lingual summaries for several languages. Chunks generated by the syntactic pattern and the Stanford CoreNLP helped our approach to generate more informative compressions, which results in better ROUGE scores.

As expected, English translations have fewer words and a smaller vocabulary (difference bigger than 1000 words) than source texts because of morphological differences between languages (Table 5.12). These translations also have shorter sentences and a

Methods	French			Portuguese			Spanish		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
baseline.late	0.4149	0.1030	0.1594	0.4161	0.1010	0.1576	0.4107	0.1083	0.1603
baseline.early	0.4163	0.1021	0.1602	0.4135	0.1003	0.1580	0.4148	0.1132	0.1644
SimFusion	0.4179	0.1042	0.1607	0.4157	0.0999	0.1582	0.4099	0.1103	0.1616
CoRank	0.4645	0.1326	0.1939	0.4715	0.1415	0.2015	0.4602	0.1414	0.1966
Our approach	**0.4727**	0.1375	**0.1969**	**0.4743**	**0.1466**	**0.2047**	**0.4725**	**0.1458**	**0.2027**
Our approach w/o chunks	0.4704	**0.1391**	0.1963	0.4731	0.1444	0.2037	0.4648	0.1393	0.1975

Table 5.11: *ROUGE F-scores for cross-lingual summaries from French, Portuguese, and Spanish languages to English language.*

more stable vocabulary size than French translations and source texts. The combination of syntactic patterns and the Stanford CoreNLP led to the same characteristics of chunks in terms of numbers and sizes.

Characteristics	French		Portuguese		Spanish	
	Source	En-Transl.	Source	En-Transl.	Source	En-Transl.
No. words	41071	35929	37339	35244	40440	37066
No. vocabulary (tokens)	8837	7718	8694	7615	8808	7703
No. sentences	2000	2000	2002	2002	1787	1787
Sentence length	20.5	18.0	18.6	17.6	22.6	20.7
No. chunks	–	4302	–	4327	–	4324
Average length of chunks	–	2.3	–	2.3	–	2.3

Table 5.12: *Statistics of datasets and their translation to English.*

Table 5.13 details the clustering and the compression processes for the English translations. These translations from French source texts have more clusters because we used a smaller similarity threshold to consider two sentences as similar. English summaries from French have more compressions in the summaries because of the large number of clusters.

French and Portuguese source texts have almost the same number of sentences, while English and Spanish source texts has fewer sentences. MT has a major role in CLTS because it generates a new specific vocabulary from French text documents. This new vocabulary can be larger or smaller than the vocabulary of the source language, which changes values of similarities between the sentences. These changes modify the clustering of sentences and the results of the CoRank method. For example, the French translations generated from English have more clusters and lower compression ratios than other languages.

Comparing the results of English and French translations, English compressions are

Characteristics	French	Portuguese	Spanish
No. clusters	128	69	84
Average size of clusters	2.7	2.7	2.8
Average length of clusters	22.1	19.2	27.0
Average length of compressions	16.4	16.3	21.1
Average number of compressions in summaries	2.5	0.9	1.5
Average compression ratio of compressions	74.2%	84.9%	78.1%

Table 5.13: *Statistics about clusters and compressions for English translated texts.*

shorter than French compressions. The use of chunks in MSC improved the results of our cross-lingual summaries, especially for English translations that have chunks that are more numerous and complex than French translations. The threshold plays an important role in clustering similar sentences and in removing redundant sentences. The use of an adaptable threshold for each language may improve the quality of the clustering and the summary generation.

Unfortunately, a manual evaluation is inviable because of the huge amount of cross-lingual summaries for all pairs of languages. However, the manual evaluation of our French-to-English cross-lingual summaries (Section 5.1.2) showed a correlation between ROUGE-1 and the informativeness scores, i.e. summaries with better ROUGE-1 achieved better informativeness scores. Therefore, we consider our cross-lingual summaries are more informative than extractive summaries for all pairs of languages.

To sum up, our approach has shown to be more stable than CoRank, thus generating more informative cross-lingual summaries with consistent ROUGE scores measured in several languages.

5.2.4 Conclusion

Cross-Language Text Summarization (CLTS) produces a summary in a target language from documents written in a source language. It implies a combination of the processes of automatic summarization and machine translation. Unfortunately, this combination produces errors, thereby reducing the quality of summaries. Joint analysis allows CLTS systems to extract relevant information from source and target languages, which improves the generation of extractive cross-lingual summaries. Recent methods have proposed compressive and abstractive approaches for CLTS; however, these methods use frameworks or tools that are available in only a few languages, limiting the adaptability of these methods to other languages. Our Multi-Sentence Compression (MSC) approach generates informative compressions from several perspectives (translations from different languages) and achieves stable ROUGE results for all languages. In addition, our method can be easily adapted for other languages.

Chapter 6

Cross-Language Text Summarization Applications

Contents

In order to analyze and to develop Cross-Language Text Summarization (CLTS) systems for complex problems, we participated in the MC2 CLEF Lab and the European Chistera AMIS project. The MC2 CLEF Lab aims to contextualize microblogs by generating small descriptions of their subjects in several languages. This contextualization combines information retrieval to identify Wikipedia pages similar to a microblog, and CLTS to produce a short description of these pages in several languages. We developed a system to contextualize microblogs by splitting this task into a pipeline consisting of three main parts: Information Retrieval (IR), Text Summarization (TS) and Machine Translation (MT) (Linhares Pontes et al. (2017, 2018a), Section 6.1).

The European Chistera AMIS project aims to generate a video summary with information in a target language from video in a source language. This project combines

several tasks: speech recognition, machine translation, text and video summarization. In this context, we extended our CLTS system (Chapter 5) to generate cross-lingual summaries of transcript documents (Linhares Pontes et al. (2019), Section 6.2). The analysis of transcript documents is particularly challenging for CLTS which has to deal with speech recognition errors and the characteristics of oral language.

6.1 Microblog Contextualization

The MC2 CLEF 2017[1] Lab analyzed the context and the social impact of microblogs (Jones et al., 2017). This Lab was composed of three main tasks: 1/ Content Analysis, 2/ Microblog Search, and 3/ Time Line Illustration. The Content Analysis task involved itself several items: classification, filtering, language recognition, localization, entity extraction, linking open data, and summarization of Wikipedia pages and microblogs. Specifically, the summarization item, on which we focus here, aims to generate a textual summary using Wikipedia pages to contextualize a microblog in four languages (English, French, Portuguese, and Spanish).

In this section, we present the challenges of the MC2 task to contextualize microblogs in four languages. We describe the system of our last year's participation in this Lab (Linhares Pontes et al., 2017). Unfortunately, the organizers of MC2 CLEF 2017 did not provide the results of the microblog contextualization task. Therefore, we only provide here an analysis of the advantages and limitations of our approach. Our system extracts information from several language versions of Wikipedia to contextualize microblogs by generating cross-lingual summaries of Wikipedia pages. Our approach analyzes this task in several subtasks, each being prone to errors. This requires to measure how each subtask acts on the quality of summaries. Therefore, we propose an evaluation protocol to evaluate this task in two ways: end-to-end and by subtask.

This section is organized as follows. Section 6.1.1 briefly describes a baseline approach and an overview of the architecture to tackle the MC2 task. Next, in Sections 6.1.2 and 6.1.3, we analyze the challenges of this task, the advantages and limitations of our approach. Then, we propose a protocol to evaluate this task in several ways in Section 6.1.4. Finally, we make final conclusions in Section 6.1.5.

6.1.1 System Architecture

A simple baseline for the MC2 task aims to retrieve information about a festival in a microblog from the Wikipedia databases in four languages (English, French, Portuguese, and Spanish). Then, the baseline selects the most relevant sentences that describe this festival to generate a short summary of 120 words independently for each language version. However, this approach does not cross-check the facts between languages and

[1] https://mc2.talne.eu/

an extractive summarization may contain several irrelevant words that reduce the informativeness of summaries.

In order to improve informativeness, we jointly take into account several language versions of Wikipedia and the sentences are compressed in order to retain only the relevant information. However, this consideration increases the complexity of the MC2 task. Considering these problems, we proposed a system that split this task into subtasks. In this regard, we present their challenges, advantages, and limitations.

Our system is composed of two main parts. The first one (Figure 6.1, left side) aims to retrieve the Wikipedia pages that best describe the festival mentioned in a microblog (Section 6.1.2). Then, we scored these Wikipedia pages according to their relevance with respect to a microblog.

The second part (Figure 6.1, right side) analyzes the best scored pages, then it extracts the relevant information from this subset in order to generate a short text summary. Our approach creates clusters of similar sentences, then we use a CLTS system (Section 6.1.3) to compress the clusters and then generate summaries in four languages describing a festival.

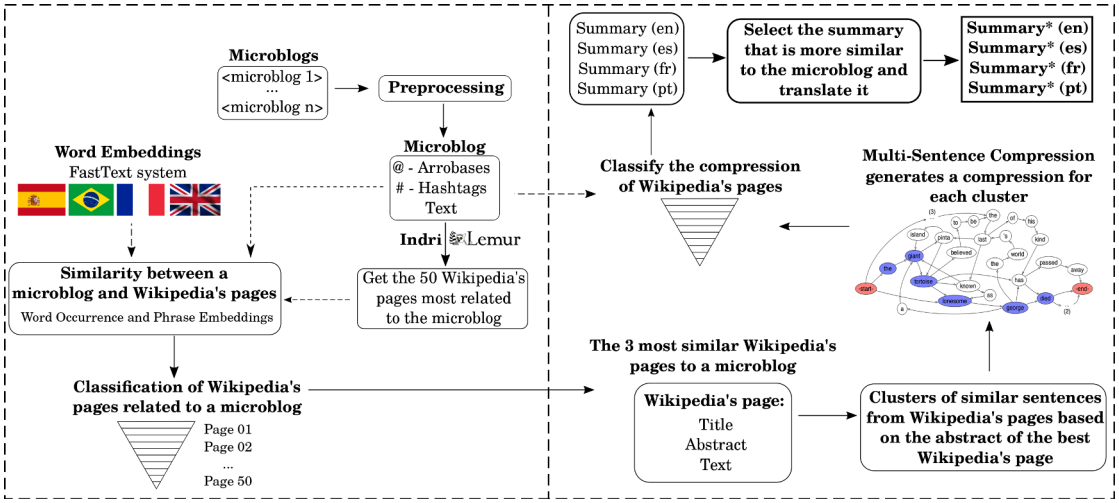


Figure 6.1: *Our system architecture to contextualize microblogs.*

6.1.2 Wikipedia Document Retrieval

The set of CLEF microblogs is composed of tweets in several languages related to festivals around the world. Wikipedia provides a description of a given festival in several languages (e.g. the Avignon Festival has a dedicated page in 17 languages). We independently analyze four language versions of Wikipedia (en, es, fr, and pt) for each microblog, by repeating the whole process first to retrieve the best Wikipedia pages and then to summarize the pages for the four versions of Wikipedia.

The following subsections describe the procedure to analyze and to retrieve the Wikipedia pages which are the most related to a festival in a microblog.

Wikipedia Page Retrieval

The first challenge of the MC2 task is to retrieve the Wikipedia pages that best describe a festival in a microblog. A microblog is written in a specific language and contains usernames, hashtags, text, and punctuation marks. Based on this microblog, a system has to identify the most relevant Wikipedia pages in four languages with respect to a festival.

We assume that hashtags and usernames represent the keywords of a tweet, and they are independent of the language. In other words, the festival name, its geographic localization, or a show name normally have the same name in different languages (e.g. "Festival d'Avignon" in French and "Avignon Festival" in English share the same keywords). We remove all punctuation marks. From hashtags, usernames, and the plain text (i.e. the tweet without hashtags, usernames, and punctuation), we create Indri queries to retrieve 50 Wikipedia documents per each microblog[2]. These Indri queries have hashtags, usernames, and the word "festival" as keywords and the retrieved Wikipedia pages must contain at least one of these keywords (Linhares Pontes et al., 2016).

The procedure described above is simple but has several limitations. Some language versions of the Wikipedia database have very little information or no page at all about a festival. In this case, the Indri system may retrieve pages about other festivals (e.g. "Avignon Festival" is not available in Portuguese). Besides, some of these festivals have names that vary according to the language and our system does not translate these names to retrieve these pages in other languages. Another characteristic that we do not take into account is the date of a microblog. Normally, people write their microblogs during festivals, therefore timestamp could have helped us to identify the correct festival.

Selection of Wikipedia Pages

The Wikipedia pages retrieved by the Indri system may contain several subjects. Indri returns these pages sorted by relevance, where the first page is the most relevant, the second is less relevant and so on. However, the quality of these results depends on the Indri query and the amount of information available about a festival. Some microblogs only contain limited information about a festival, e.g. the location of a festival or the name of a show. In this case, a system has to identify the correct festival among several with similar characteristics, presentations in common, or in the same location.

To confirm the relevance of the Wikipedia pages retrieved by Indri, we select the pages most related to a microblog. Normally, the title of a Wikipedia document has few

[2] `https://www.lemurproject.org/indri.php`

words and contains the core information, while the abstract of the document, which is usually made of the first paragraphs of the article before the start of the first section, is larger and provide additional information[3]. Therefore, we consider Equation (6.4) to compute the relevance score of the Wikipedia document D with respect to the microblog T.

$$\text{score}_{title} = \alpha_1 \times \text{sim}_{mb}(ht, title) + \alpha_2 \times \text{sim}_{mb}(un, title) + \alpha_3 \times \text{sim}_{mb}(nw, title) \quad (6.1)$$

$$\text{score}_{abs} = \beta_1 \times \text{sim}_{mb}(ht, abs) + \beta_2 \times \text{sim}_{mb}(un, abs) + \beta_3 \times \text{sim}_{mb}(nw, abs) \quad (6.2)$$

$$\text{sim}_{mb}(i, j) = \gamma_1 \times \text{cosine}(i, j) + \gamma_2 \times \text{occur}(i, j) \quad (6.3)$$

$$\text{score}_{doc} = \text{score}_{title} + \text{score}_{summary} \quad (6.4)$$

where ht are the hashtags of the tweet T, un the usernames of T, nw the normal words of T, and abs the abstract of D. $\text{occur}(i, j)$ represents the number of occurrences of i in j, while $\text{cosine}(i, j)$ is the cosine similarity between i and j using Continuous Space Vectors[4] (Bojanowski et al., 2017c).

We empirically set up the parameters as follows: $\alpha_1 = \alpha_2 = 0.1, \alpha_3 = 0.01, \beta_1 = \beta_2 = 0.05, \beta_3 = 0.005, \gamma_1 = 1$ and $\gamma_2 = 0.5$. These coefficients give more weights to hashtags than usernames and the tweet text, and compensate the shorter length of the titles of Wikipedia articles with respect to their abstract. These pages may contain several subjects and we only want to keep the pages that describe the festival of the microblog. Therefore, we finally keep in each language the three Wikipedia documents with the highest scores to be analyzed by the TS system.

Our system prioritizes the information in hashtags written with a # symbol and in usernames starting with an @ sign; however, a microblog has few information about a festival and, sometimes, this information is too general or too specific to easily identify a festival. Another problem is that the Wikipedia dataset has several kinds of pages, e.g. lists of festivals based on a show, cities, or types of festival. These pages contain irrelevant information about a particular festival and may reduce the informativeness of summaries.

6.1.3 Text Summarization

One of the biggest challenges of the Microblog Contextualization task is to summarize all the information available in a correct and informative summary about a festival. As described earlier, the retrieved pages may contain wrong information because they may be in different languages and describe various festivals.

[3]We did not consider the whole text of Wikipedia pages because it is sometimes huge and we preferred to rely on the work of the contributors to build the summary of the article.

[4]We used the pre-trained word embeddings (en, es, fr, and pt) of FastText system (Bojanowski et al., 2017c) that is available in `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`.

While famous festivals have several Wikipedia pages that describe in detail all previous editions, less prominent ones have only one page or no article at all in Wikipedia. For this reason, we use the best scored page as the reference for the contextualization of microblogs. This analysis helps to have access to the correct subject and avoid using information about other subjects. The abstract provided at the start of the Wikipedia pages is assumed to be good enough to be coherent and to provide a basic description of a festival. However, relying only on this part of the article may lead to miss relevant information about the festival that could be obtained from other sections or even other pages. For this reason, we preferred to use the abstract of the top article as a basic summary and to improve its quality with relevant information using MSC (i.e. generate sentences that are shorter and more informative than the original sentences of the document). Then, we translate the best summaries for the languages that have poor summaries.

In the case some Wikipedia pages - in principle, short articles - do not have an abstract, the whole text is analyzed. This text may have additional information that is not relevant to contextualize a festival in only 120 words. Therefore, our approach strongly depends on the best scored page abstract to generate a correct summary.

Clustering

Clustering enables the identification of subjects and relevant information inside a document. These clusters are composed of similar sentences[5]. The objective of this process is to divide a document in topics where each cluster describes a specific topic.

As we consider the sentences of the abstract of the best scored page as key sentences, we create clusters made of sentences from the first three retrieved pages and similar to each key sentence. Two sentences are considered as similar if the cosine similarity between them is bigger than a threshold[6].

It can happen that some festivals have only a single relevant Wikipedia page. The cosine similarity normally helps in selecting only pertinent sentences; however, particularly in this case, sentences which are similar to key sentences may deal with different subjects and may still be included in clusters with irrelevant information.

Multi-Sentence Compression

The problematic of text summarization is to produce summaries that are both grammatical and informative while meeting length restrictions, 120 words in the task considered here. Since most of the sentences in Wikipedia are long, we attempt to compress them to preserve only the relevant information. We use our MSC method presented in Chapter 4 to generate a shorter and hopefully more informative compression for each cluster. Like in (Linhares Pontes et al., 2016), keywords have a fixed relevance of 0.9.

[5]We used the NLTK library to perform the sentence segmentation.

[6]We empirically set up a threshold of 0.4 to consider two sentences as similar.

Our approach assumes that clusters are composed of only correct sentences (subject+verb+object) to generate correct compressions. Another limitation is the similarity of sentences in a cluster. A cluster has to describe a single topic; otherwise, MSC will merge information of several subjects and generate a compression with wrong information.

Summary Generation

The last step of summarization is the generation of summaries. While original sentences are likely to be more grammatically correct than compressions, the compressed sentences are by definition shorter and have in principle more relevant information. Therefore, we prefer to add a compression in the summary if this compression is considered more relevant than the original sentences.

We generate summaries by concatenating the most similar compressions to a microblog without redundant sentences. The relevance of sentences/compressions is calculated based on the average TF-IDF. We add a sentence/compression to the summary only if the cosine similarity between this compression and the sentences already added in the summary is lower than a threshold of 0.4.

Let us note that our approach does not check the time of facts and consequently, it may generate summaries that do not preserve the sequence of facts.

Best Summary

The best possible scenario is the generation of a summary for each language version of Wikipedia. However, some language versions do not have a page or have a small text describing a specific festival. Therefore, we analyzed four summaries (one for each language version of the Wikipedia) for each microblog and we only retain the summary which contains the description most similar to the microblog. We consider a summary as relevant if it is similar to the microblog. As the translation process generates some errors, we translate a language version summary only if the quality of the best summary is much better than other versions[7]. In such case, we used the Yandex library[8] to translate the kept summary into other languages (en, es, fr, and pt).

The pipeline made of the summarization and translation processes is prone to errors, which reduces the quality of summaries. However, we have to use information from other language versions of Wikipedia when the available information about a festival in a language is poor or does not exist.

[7] We translate a summary into a target language only if the summary in the target language has a similarity score (cosine similarity between the summary and the microblog) lower by 0.2 than the similarity score between the best summary and the microblog.

[8] `https://tech.yandex.com/translate/`

6.1.4 Proposed Evaluation Protocol

INEX organizers have proposed the LogSim measure[9] to evaluate informativeness of produced contexts or summaries (Bellot et al., 2016). However, the MC2 CLEF 2017 task contains several subtasks to contextualize a large amount of microblogs in several languages. The automatic evaluation of this task as an end-to-end problem generates incomplete results. Unfortunately, MC2 CLEF 2017 organizers did not provide the results of the microblog contextulization task. In our opinion, the best way to evaluate this task is to split it in two subtasks (Wikipedia page retrieval and Text Summarization (TS)). In this case, we can estimate the impact of each subtask in the contextualization.

Our proposition for the evaluation protocol is composed of three steps: Wikipedia page retrieval, TS and the combination of the previous steps (Figure 6.2). For the Wikipedia pages retrieval subtask, systems have to determine which Wikipedia pages describe a festival in a microblog. The TS subtask consists in generating a summary of a festival based on one or several Wikipedia pages. Finally, the microblog contextualization task is composed of both subtasks.

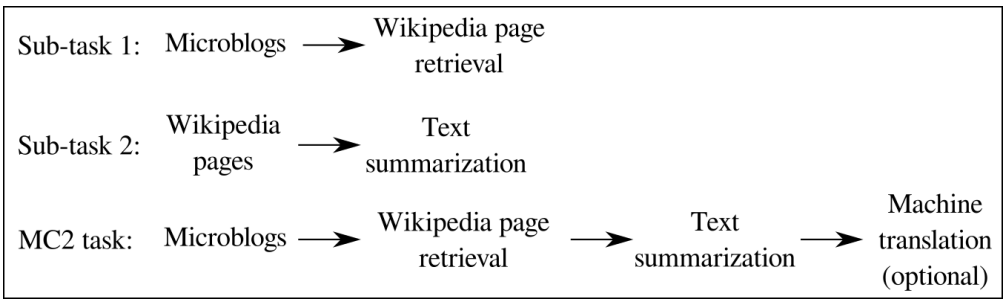


Figure 6.2: *Proposition of an evaluation protocol for MC2 task composed of two subtasks.*

The Wikipedia page retrieval subtask can be evaluated with a list of the Wikipedia pages related to a microblog. However, this list has to be annotated by human annotators. This list has to be created by the human annotator to assure the quality of annotations. The TS subtask and microblog contextualization task can be analyzed in several ways: automatic, semi-automatic and manual evaluations. Automatic (FRESA (Torres-Moreno, 2014)) and semi-automatic (ROUGE (Lin, 2004)) evaluation systems analyze the overlap of n-grams between reference summaries and original text (FRESA), and between reference summaries and candidate summaries (ROUGE) to determine the quality of candidate summaries. However, compression and translation methods change the structure of sentences by generating paraphrases and new n-grams that may not exist in reference summaries (or source document), thereby reducing ROUGE (or FRESA) scores. In this case, a manual evaluation is required to evaluate the quality of these summaries.

[9] LogSim combines the ideas of ROUGE system and Kullback-Leibler divergence. This measure compares the frequency distributions between a sample of reference passages from a very large collection of documents and the summaries of these documents.

6.1.5 Conclusion

The Microblog Contextualization task is composed of several challenges that can modify the quality of results. Depending on the microblog, this task may require the generation of multi-lingual and cross-lingual summaries. Our system is modular and can contextualize microblogs with several approaches. For example, we can remove MSC and/or the automatic translation methods in our approach. However, since this task involves several subtasks, the performance of our system depends on all these subtasks. This pipeline of subtasks complicates the identification of errors and the performance analysis of our approach. Another major problem is the lack of a training corpus to test and to adapt our system for this task. With this dataset, we could evaluate and improve our system. We also want to adapt the idea proposed in (Sadat et al., 2002) to improve the performance of our information retrieval method to recuperate cross-lingual information from Wikipedia pages in several languages.

The CLTS methods described in Section 2.6 need a group of documents that describe a same subject to generate a correct summary; however, the MC2 task does not necessarily provide correct documents about a festival and the use of these methods can generate poor summaries. A possible solution is to ensure the quality of the source documents about a same subject and to adapt these methods to analyze Wikipedia pages.

Next section describes an adaptation of our CLTS approach (Chapter 5) to generate cross-lingual summaries of transcript documents.

6.2 Cross-Language Speech-to-Text Summarization

Nowadays, audio data are part of daily life in the form of news, interviews and conversations, whether it is on the radio or on the Internet. A manual analysis of these data would be impossible because it requires a huge number of persons to analyze this information in the time available. One way to analyze and accelerate the data processing is Automatic Speech Summarization, which differs from the traditional Automatic Text Summarization task (Torres-Moreno, 2014) because there are other problems to take into account like speech recognition errors, the lack of sentence boundaries, the wide range of sentence sizes, colloquialisms and uneven information distributions (Furui et al., 2004; Christensen et al., 2003; Taskiran et al., 2006);(Linhares Pontes et al., 2015). Formally, Speech-to-Text Summarization consists of generating a short summary of the transcript documents.

We tested our French-to-English CLTS framework using MSC method (Section 5.2) to summarize transcript documents (Linhares Pontes et al., 2019). In a nutshell, we add Automatic Speech Recognition (ASR) errors in the MultiLing Pilot Dataset and we combine Automatic Segmentation method and our approach to summarize this dataset.

The rest of this section is organized as follows: we first describe the project related to this work (Section 6.2.1). We make an overview of relevant works for speech-to-text summarization in Section 6.2.2. The experimental setup and results are presented in

Sections 6.2.3 and 6.2.5, respectively. Finally, we give our conclusion and some last remarks in Section 6.2.6.

6.2.1 Access Multilingual Information opinionS (AMIS)

AMIS is a Chist-Era project[10] with the collaboration of the University of Lorraine (France), AGH University (Poland), University of Deusto (Spain) and University of Avignon (France). This project concerns human language understanding and grounding language learning. The main objective of AMIS is to make available a system, helping people to understand the content of a source video by presenting its main ideas in a target understandable language. One of the possibilities to reach this objective is to summarize the amount of information and then to translate it into the end-user language (Smaïli et al., 2019; Grega et al., 2019). This project integrates several systems (video summarization, automatic speech recognition, machine translation, language modeling, text summarization and so on) to generate cross-lingual video summaries of newscast and reports.

6.2.2 Related Work

Speech-to-text summarization has to face three main problems: documents are not segmented into sentences, they may contain speech disfluencies[11], specific to the oral language, or they are subject to misrecognized words when using ASR. Nevertheless, it can benefit from acoustic and prosodic cues, or information about the role of speakers to determine the importance or the structure of an utterance. McKeown et al. (2005) showed how the summarization approaches used in TS can be adapted to this speech-to-text task. They focused on two types of spoken sources, broadcast news and meetings, by taking advantage of acoustic, prosodic, lexical, and structural features to detect speakers' turns and overcome the difficulties that are present in spoken language.

Mrozinski et al. (2006) applied an extractive summarization approach over broadcast news stories and conference lectures. In a first step, they performed sentence segmentation of the transcripts using word-based and class-based statistical language models; then during the summarization phase they selected the highest scoring sentences based on a combination of word significance score, confidence score, and linguistic likelihood.

Rott and Červa (2016) divided their summarization system in three steps: automatic speech recognizer, syntactic analyzer, and text summarizer. Sentence Boundary Detection (SBD) was performed during the syntactic analysis, where they identified phrases in the recognized text using syntactic engineering tool (Kovář et al., 2009). Text summarization was performed using a TF-IDF method which selects the most informative phrases.

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

[11] A speech disfluency is any disruption in the flow of spoken language that is caused by the speaker, e.g. stuttering and hesitations.

6.2.3 Experimental Setup

We compare our method (Compressive Cross-Language Speech-to-Text Summarization (CCLSTS)) with the early translation, the late translation, SimFusion and CoRank (Wan, 2011). The early and late translations are based on the SimFusion method, the difference between the systems is that similarity metrics are computed either in the target language (early translation) or in the source language (late translation).

We only compress sentences with more than 10 words to avoid short outputs with little information. The MSC method selects the 10 most relevant keywords per topic and the 3 most relevant keywords per cluster of similar sentences to guide the compression generation. All analyzed systems generate summaries composed of 250 words with the most relevant sentences, while the redundant sentences are discarded. We apply the cosine similarity measure with a threshold θ of 0.5 to create clusters of similar sentences for MSC and to remove redundant sentences in the summary generation.

6.2.4 Dataset

We used the MultiLing Pilot 2011 dataset (Section 5.1.2). This dataset is composed of 10 topics, each topic having 10 source texts and 3 reference summaries. Specifically, we use the French version of the MultiLing Pilot 2011 dataset as source language and the corresponding English version as the target language.

To our knowledge, no work has been done regarding cross-language summarization of transcripts generated by an Automatic Speech Recognition (ASR) system. We believe this to be a good challenge given the difficulties brought by ASR transcripts. For this reason we wanted to explore this less controlled scenario and analyze the repercussions over the cross-language text summarization of two main problems of ASR transcripts: transcription errors and the lack of sentences.

Transcription error simulation

Automatic transcription performance is normally compared against one or more references using Word Error Rate (WER). 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 (Equation 6.5) are deletions, insertions and substitutions:

$$\text{WER} = \frac{DEL + INS + SUB}{n} \tag{6.5}$$

where DEL corresponds to the number of deletions, INS to the number of insertions, SUB to the number of substitutions and n to the number of words in the reference. An ASR transcript carries all three errors at different ratios; for this controlled scenario

we simulated in an isolated way each error to observe how each of them affects the performance of cross-language speech-to-text summarization individually.

We approximated WER by simulating the errors produced by ASR systems in a straightforward approach. The deletion error dataset (ASR_DEL) was created by choosing m words of each document randomly and by deleting them. Concerning the substitution error dataset (ASR_SUB), for each document we first selected a set $R = \{r_1, ..., r_m\}$ of words randomly, then for each word w_i of the document a randomly generated decision value $v_i \in [0, 1]$ was calculated; if v_i happened to be greater than a given threshold 0.5, then w_i was replaced by r_j, this cycle was repeated until all words r_j in R were picked. The insertion error dataset (ASR_INS) followed the same procedure as ASR_DEL but instead of replacing w_i by r_j, r_j was placed after w_i. Finally, the ASR_MIX dataset is composed of a combination of insertion, deletion and substitution errors. For all four error datasets m was calculated as:

$$m = \text{WER} \times n \tag{6.6}$$

where n corresponds to the length (number of words) in each original document and WER was fixed to 0.15[12].

Automatic Segmentation

We are aware that this simulation of ASR errors is not realist, but accurate realistic models are very complex to develop and never satisfactory. Notably, the types of errors can be variable according to the ASR system used. Common ASR transcripts have no punctuation, which further complicates NLP tasks like automatic summarization. We simulated the lack of punctuation by deleting all punctuation signs inside the MultiLing Pilot 2011 French dataset (ASR_NO) and the datasets with induced transcription errors (ASR_DEL, ASR_SUB, ASR_INS, ASR_MIX); then we automatically restored them. This task is known as SBD.

To restore the punctuation within the corpus, we followed the best model reported by González-Gallardo and Torres-Moreno (2018). This approach formulates the segmentation problem as a classification one. It uses a CNN with subword-level information vectors (Bojanowski et al., 2017c) to predict if the centered word (w_i) within a window $\{w_{i-(m-1)/2}, ..., w_{i-1}, w_i, w_{i+1}, ..., w_{i+(m-1)/2}\}$ corresponds to a sentence border or not.

The hidden architecture of the CNN consists of two convolutional layers with a valid padding and a stride value of one, followed by a max pooling layer and three fully connected layers with a dropout layer attached at the end. The outputs of all convolutions, max pooling and fully connected layers have a RELU activation function. We used the French pre-trained subword-level information vectors in dimension

[12]The performance of ASR systems depends on the type of dataset. We set up the WER of 0.15, which is the average WER of recent ASR approaches (Xiong et al., 2016; Fohr et al., 2017).

Datasets	Class	Precision	Recall	F-score
ASR_NO	NO_BOUND	**0.971**	**0.986**	**0.978**
	BOUND	**0.840**	**0.721**	**0.776**
ASR_DEL	NO_BOUND	0.966	0.963	0.965
	BOUND	0.654	0.673	0.663
ASR_INS	NO_BOUND	0.960	0.956	0.958
	BOUND	0.592	0.616	0.604
ASR_SUB	NO_BOUND	0.958	0.950	0.954
	BOUND	0.554	0.600	0.576
ASR_MIX	NO_BOUND	0.963	0.958	0.960
	BOUND	0.614	0.643	0.629

Table 6.1: *Results of Sentence Boundary Detection over the Automatic Speech Recognition datasets.*

300 trained on Wikipedia using fastText[13]. The CNN was trained with on 380M-word corpus derived from the French Wikipedia.

Table 6.1 presents the automatic evaluation performed over the unpunctuated datasets. As seen from the “no boundary" class (NO_BOUND), the method has a really good performance (over 0.95 for all metrics), regardless of the type of transcription errors. Given the unbalanced nature of the data this is an expected behavior. Nevertheless, for the “boundary" class (BOUND) the performance drops when trying to segment the noisy transcripts. The worst scenario corresponds to the dataset with substitution errors (ASR_SUB), where precision and recall present relative drops of 34% and 17% against ASR_NO.

Automatic Text Summarization Evaluation

Automatic Text Summarization Evaluation relies on comparing the information contained in the generated (candidate) summary against one or more reference summaries or the source document. Therefore, we considered ROUGE-1, ROUGE-2 and ROUGE-SU4 to evaluate and compare our system (more details about the ROUGE measure in Section 4.3.2).

6.2.5 Experimental Evaluation

Table 6.2 shows the ROUGE scores for each version of the MultiLing Pilot dataset. Our method outperformed the other methods for the original, ASR_NO, and ASR_SUB dataset versions, while the CoRank method obtained the best results for ASR_INS and

[13] `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`

ASR_DEL dataset versions. As we expected, the ASR errors, introduced at the word or segmentation levels, reduced the performance of systems.

We analyzed the original dataset results as a reference to compare the performance of the systems with other dataset versions. The joint analysis of both languages generated better results. The analysis of the similarity in both languages and cross-language increased the results considerably. Finally, the addition of the compression of similar sentences to these multiple analysis of similarities achieved the best results.

The automatic segmentation process may split long sentences in two or more short sentences that can be more or less relevant to the document. In addition, these sentences are more likely to contain grammatical errors. However, the segmentation errors had little impact on the performance of systems (ASR_NO in tables 6.1 and 6.2).

The low performance of automatic segmentation process to identify sentence boundaries combined with ASR errors reduced the performance of all systems (ASR_DEL, ASR_INS, ASR_SUB and ASR_MIX in Tables 6.1 and 6.2). These errors modified the structure of sentences causing large translation errors and changing the meaning of some sentences. Surprisingly, early translation, late translation, and SimFusion improved their ROUGE scores on the ASR_MIX dataset. The combination of types of errors modified the sentences and their similarity values which improved the performance of these methods. The CoRank method achieved the best results for the deletion and insertion dataset versions; however, poor results were obtained for the substitution errors.

Documents with ASR errors normally have sequences of words with unusual co-occurrence of words. The analysis of cohesion of words in our MSC method identified the passages of the documents with low cohesion (infrequent co-occurrence of words) and generated compressions by avoiding these passages. Although we did not achieve the best results for all datasets, our approach was more stable for all kinds of ASR errors by generating cross-lingual summaries with similar ROUGE scores.

To sum up, the joint analysis of information in both languages and MSC generates more informative cross-lingual summaries. Our segmentation process kept a good quality of all summaries, i.e. all systems generated summaries with ROUGE scores similar to the original dataset. The addition of ASR errors reduced the quality of summaries of all systems because of translation and meaning errors. Our approach generated cross-lingual summaries with similar ROUGE scores for the dataset with ASR errors while the CoRank method achieved unstable results depending on the kinds of error.

6.2.6 Conclusion

We have proposed a compressive method to improve the generation of cross-lingual transcript summaries. The addition of the automatic segmentation method in our CLTS approach (Section 5.2) allowed the segmentation of transcription documents with several types of errors. This approach analyzes transcription documents in both languages

Dataset	Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
Original	Early translation	0.4141	0.1025	0.1594
	Late translation	0.4115	0.1034	0.1581
	SimFusion	0.4149	0.1046	0.1596
	CoRank	0.4660	0.1343	0.1953
	CCLSTS	**0.4761**	**0.1373**	**0.2005**
ASR_NO	Early translation	0.4147	0.0979	0.1584
	Late translation	0.4113	0.0990	0.1567
	SimFusion	0.4163	0.0988	0.1598
	CoRank	0.4603	0.1253	0.1888
	CCLSTS	**0.4726**	**0.1434**	**0.1997**
ASR_DEL	Early translation	0.4145	0.0944	0.1562
	Late translation	0.4072	0.0896	0.1503
	SimFusion	0.4128	0.0904	0.1539
	CoRank	**0.4580**	**0.1135**	**0.1799**
	CCLSTS	0.4417	0.1050	0.1718
ASR_INS	Early translation	0.4021	0.0861	0.1492
	Late translation	0.3950	0.0849	0.1444
	SimFusion	0.3988	0.0848	0.1467
	CoRank	**0.4508**	**0.1092**	**0.1773**
	CCLSTS	0.4385	0.1059	0.1715
ASR_SUB	Early translation	0.4101	0.0846	0.1510
	Late translation	0.4051	0.0842	0.1471
	SimFusion	0.4074	0.0845	0.1498
	CoRank	0.4270	0.0939	0.1606
	CCLSTS	**0.4412**	**0.0964**	**0.1691**
ASR_MIX	Early translation	0.4327	0.0965	0.1609
	Late translation	0.4345	0.0941	0.1608
	SimFusion	0.4325	0.0971	0.1610
	CoRank	0.4531	**0.1098**	**0.1783**
	CCLSTS	**0.4534**	0.1085	0.1779

Table 6.2: *ROUGE F-scores for French-to-English cross-lingual summaries using MultiPilot 2011 dataset.*

to identify relevant information and compress similar sentences to increase the informativity of cross-lingual transcript summaries. The simulated ASR errors showed to have an impact on the performance of all systems; nevertheless, our approach achieved the best results for the original, ASR_NO and ASR_SUB dataset versions. Contrary to the CoRank method, our approach attained stable results for all kinds of ASR errors.

Actual transcript documents contain more challenges than our simulation of ASR errors. In addition to the lack of punctuation marks and the existence errors of word recognition, transcript documents are composed of sentences that have different lengths, vocabularies and colloquialisms. These additional problems complicate the sentence segmentation and the analysis of sentences to compress them. In order to mitigate the accumulation of errors generated by the combination of ASR and MSC, we will consider the grammatical quality of the sentences to compress only sentences with a correct syntactic structure. We will also use a language model or neural networks to correct grammatical errors (Yuan and Briscoe, 2016) generated by ASR in order to improve the quality of transcripts and, consequently, the quality of summaries.

Chapter 7

Conclusion and Future Work

Contents

7.1 Conclusion

The huge amount of information available on the Internet has facilitated to be up to date on world news. However, all this information cannot be read in a feasible time. Therefore, Text Summarization (TS) systems are useful to analyze and to select which information can represent the gist of all these data. This thesis split this problem in several subtasks in order to build a more efficient compressive Cross-Language Text Summarization (CLTS) system.

The first subtask was to predict the semantic similarity of two sentences. The order of words in a sentence changes the meaning of a sentence; therefore, the analysis of a sentence as a bag of words does not consider the meaning of multi-word expressions, which reduces the capacity of similarity prediction between two sentences. In order to consider the order of words and their multi-word expressions in the sentence analysis, we combined CNN and LSTM structures to analyze, identify and preserve the relevant information in each part of sentences and in the whole sentences. The local context carried out a more in-depth sentence analysis by providing complement information about the word in the sentences. In our experiments, the local context improved the prediction of the sentence similarity, by reducing the mean squared error and increasing the correlation scores. Unfortunately, the training corpus (SICK dataset) for the semantic similarity is small, which reduces the performance of our system to analyze the similarity of sentences that do not share the same context of the training corpora. For example, the prediction of the similarity of news sentences are more difficult because the SICK dataset does not contain such sentences. In this case, we have to train our system on larger and more general datasets that can generalize all kinds of sentences.

The second subtask was the multi-sentence compression task that aims to generate a short informative compression from several sentences with related and redundant information. We developed a new model for MSC that extends the common approach based on graph theory, using vertex-labeled graphs and integer linear programming to select the best compression. The vertex-labeled graphs are used to model clusters of similar sentences with keywords, while the optimization criterion introduces a balance to generate an informative and correct compression. Our system can generate shorter compressions with the risk to lose some information, or privilege informativeness by generating longer compressions. Evaluations led with both automatic metrics and human evaluations show that our Integer Linear Programming (ILP) model consistently generate more informative sentences than two baselines while maintaining their grammaticality for several languages. Our approach is able to choose the amount of information to keep in the compression output and to generate compressions guided by keywords. Moreover, it can be easily adapted to other languages.

Finally, we used our work on Multi-Sentence Compression (MSC) to propose a compressive method to improve the generation of cross-lingual summaries. Our system analyzes a document in both languages to extract all relevant information. Then, we apply two sentence compression methods to preserve the main information of sentences. Our MSC method generates a compression for a cluster of similar sentences. The second method uses a Neural Network (NN) model to compress sentences by deleting non-relevant words in a sentence. Our method using MSC generated more informative summaries than extractive methods. The manual evaluation proved that our approach generated more informative cross-lingual summaries without reducing the grammatical quality of summaries w.r.t. extractive approaches. Our multilingual analysis showed that our system based on MSC is more stable and is able to generate informative cross-lingual summaries for several languages. Finally, our application using CLTS for transcript texts showed to be a stable system even for texts containing several Automatic Speech Recognition (ASR) and segmentation errors.

7.2 Future Work

There are several avenues worth exploring from these works. Considering the semantic similarity, we plan to test other methods to analyze the local context (Ermakova and Mothe, 2016; Zhu et al., 2018) to compare with our CNN approach. Unfortunately, we did not find larger annotated corpora for this task to generalize all kinds of subjects and sentence constructions. Therefore, we also want to adapt the training process of our NN model to use labeled and unlabeled dataset to improve the prediction of sentences with complex syntactic structures and different subjects. Finally, we want to lead extrinsic evaluations by measuring how Semantic Text Similarity (STS) acts on TS systems.

Bilingual word embeddings provide similar representations for corresponding words in different languages. Instead of using a Siamese network, we would like to use two different LSTM+CNN models to process the context and the sentence analysis for each language. For example, the words of an English sentence require a different

analysis than those of a sentence in another language with respect to the order and meaning of the previous and following words. This cross-language sentence similarity can improve the analysis of the CoRank method and the sentence clustering process.

As regards the multi-sentence compression, we would like to manage the polysemy through the use of the same label for the synonyms of each keyword inside the Word Graph (WG). We would also like to compare the performance of polysemy in our MSC approach using external resources like WordNet to get specific information and pre-trained word embeddings to get general information about the context of the keywords in the clusters. Following the idea presented by (Nayeem et al., 2018), we will add lexical substitute words (and nodes) for nouns and verbs in WG to increase the combination of the sentences in the WG. Finally, a neural language model could be used to select the most grammatically correct compression generated by our approach.

For the cross-language text summarization, we want to extend the attention mechanism in our SC model to take into account keywords in order to guide the SC process. Tree-based and sentence-based SC approaches can be used to compress long and complex sentences in order to mitigate the poor performance of NN approaches for these types of sentences. Finally, we also want to test our semantic sentence similarity framework in CLTS to carry out an analysis of the impact of the semantic analysis in the generation of cross-lingual summaries.

In order to reduce the errors generated by the pipeline of MT and TS, we want to generate cross-language sentence compressions with an end-to-end approach. However, the lack of cross-language sentence compression datasets make this task very hard. Therefore, we want to develop a multi-task Neural Network model to combine the learning process using the information from parallel sentence translation and sentence compression. The idea is to learn this NN model in two steps. The first one calculates cross-language sentence embeddings and the second generates monolingual sentence compressions. We will combine these tasks in a same NN model to attempt to generate cross-language sentence compressions and improve the grammaticality of cross-lingual summaries.

A same language has different ways to express a same meaning, various sentence constructions and different spellings (e.g. "analyze" and "analyse") in different countries. Readers understand better a summary if this one has the characteristics of their language region. Therefore, multi-cultural aspects can be interesting in the summary generation. However, these characteristics represent a great challenge in NLP because it is very difficult to identify these differences between variations in the same language and to generate a summary using the characteristics of a specific linguistic region.

An issue with TS and CLTS applications is the evaluation procedure to determine the informativeness and the grammaticality of (cross-lingual) summaries. State-of-the-art works normally use three kinds of evaluations: automatic, semi-automatic and manual. Normally, the FRESA and the ROUGE (or Pyramid (Nenkova et al., 2007)) systems are used in the automatic and the semi-automatic evaluations, respectively. However, these evaluations only analyze the overlap of words between candidate and reference summaries (ROUGE), or between candidate summaries and source document (FRESA).

This overlap do not analyze the meaning and the grammatical analysis of summaries. The manual evaluation can analyze the informativeness and the grammaticality of summaries; however, this procedure is very slow and expensive making this analysis unfeasible for large experiments. This analysis is outside the scope of this thesis but the creation of automatic or semi-automatic evaluation systems, which can provide reliable scores for estimating the informativeness and grammar of abstracts, is fundamental for the research in NLP.

Another relevant characteristic of summaries that we did not analyze in this thesis is the readability of summaries. The connection of ideas between sentences in a summary can make a summary understandable or not. Extractive, compressive and abstractive approaches generate summaries by concatenating ideas. In most cases, this concatenation does not analyze the relationships between sentences, which reduces the comprehensibility and readability of summaries.

Appendix A

Discrete Context Vocabulary for Text Summarization

Contents

Text Summarization (TS) aims at producing a condensed text document retaining the most important information from one or more documents. Different methodologies based on graphs, optimization, word frequency or word co-occurrence have been used to automatically create summaries (Torres-Moreno, 2014). In the last years, Continuous Space Vector (CSV) has been employed in several studies to evaluate the similarity between sentences and to improve the summary quality (Balikas and Amini, 2015; Kågebäck et al., 2014; Phung and De Vine, 2015).

In this Appendix, we analyzed a complementary work on word representation for mono-lingual text summarization (Linhares Pontes et al., 2016). Unlike most works that use either a continuous or discrete representation of words, we propose a discrete context representation of words to generate a discrete representation that conserves some context information of words. In this representation, words with similar contexts have a same discrete representation. We evaluated several word representations in order to identify which representation generates more informative summaries using extractive text summarization systems.

A.1 Reduced Vocabulary

Words can be represented by two main kinds of vectors: Discrete Space Vector (DSV) and Continuous Space Vector (CSV). In DSV, words are independent and the vector dimension varies with the used vocabulary. Thus similar words (i.e., “home” and

"house", "beautiful" and "pretty") have different representations. For statistical techniques, this independence between similar words complicates the analysis of sentences with synonyms.

CSV is a more compelling approach since similar vectors have similar characteristics and the vector dimension is fixed. For CSV (word embeddings), it is possible to identify similar characteristics between words. For example, the words "home", "house" and "apartment" have the same context as "home" and have therefore similar vectors. However, the existing methods to calculate the sentence relevance are based on DSV. We use CSV to identify and replace the similar words to create a new vocabulary with a limited semantic repetition. From this reduced vocabulary, statistical techniques can identify with DSV the similar content between two sentences and improve the results.

A general and large corpus is used to build the word embedding space. Our method calculates the nearest words in this space for each word of the texts to create groups of similar words, using a cosine distance. Then it replaces each group of similar words by the most frequent word in the group. For example, the nearest word of "home" is "house" and the word "home" is more frequent than "house" in the text, so we replace the word "house" by "home". Let us note that these substitutions are only used to compute sentence similarities but that the original words are kept in the produced summary. We devised the greedy algorithm 1 to find the similar words of w in the texts among a pre-compiled list lcs of CSV generated on the large corpus.

Algorithm 1 Reduce vocabulary of *text*

Input: n (neighborhood size), lcs (list of words inside continuous space), *text*
for each word w_t in *text* **do**
  if w_t is in lcs **then**
    $nset \leftarrow \{w_t\}$
    $nlist \leftarrow [w_t]$
    while $nlist$ is not empty **do**
      $w_l \leftarrow nlist.pop(0)$
      $nw \leftarrow$ the n nearest words of w_l in lcs
      $nlist.add((nw \cap \text{vocabulary of } text) \setminus nset)$
      $nset \leftarrow nset \cup (nw \cap \text{vocabulary of } text)$
    end while
    Replace in *text* each word of *nset* by the most frequent of *nset*
  end if
end for
Return *text*

A.2 Experiments and Results

The reduced vocabulary approach was evaluated with four different systems. The first simple system (named "base") generates an extract with the sentences that are the most

similar to the document. The second system (MMR) produces a summary based on the relevance and the redundancy of the sentences (Carbonell and Goldstein, 1998). With the objective of analyzing different methodologies to calculate the relevance and the similarity of sentences (e.g. word co-occurrence, Term Frequency-Inverse Sentence Frequency (TF-ISF)...), we use two other systems: SASI (Linhares Pontes et al., 2014) and TextRank (Mihalcea and Tarau, 2004).

Linhares Pontes et al. (2014) use Graph theory to create multi-document summaries by extraction. Their so-called SASI system models a text as a graph whose vertices represent sentences and edges connect two similar sentences. Their approach employs TF-ISF to rank sentences and creates a stable set of the graph. The summary is made of the sentences belonging to this stable set.

TextRank (Mihalcea and Tarau, 2004) is an algorithm based on graphs to measure the sentence relevance. The system creates a weighted graph associated with the text. Two sentences can be seen as a process of recommendation to refer to other sentences in the text based on a shared common content. The system uses the Pagerank system to stabilize the graph. After the ranking algorithm is run on the graph, the top-ranked sentences are selected for inclusion in the summary.

We used for our experiments the 2011 MultiLing corpus (Giannakopoulos et al., 2011) to analyze the summary quality in the English and French languages. We concatenated the 10 texts of each topic to convert multiple documents into a single text. There are between 2 and 3 summaries created by humans (reference summaries) for each topic. We took the LDC Gigaword corpus (5th edition for English, 3rd edition for French) and the word2vec package[1] to create the word embedding representation, the vector dimension parameter having been set to 300. We varied the window size between 1 and 8 words to create a dictionary of word embeddings. A neighborhood of between 1 and 3 words in the continuous space was considered to reduce the vocabulary (parameter n of Algorithm 1). Finally, the summaries produced by each system have up to 100 words.

The compression ratio using the algorithm 1 depends on the number n of the nearest words used. Table A.1 reports the average compression ratio for each corpus in the word embedding space for three values of n. For the English language, a good compression happens using 1 or 2 nearest words, while the vocabulary compression for the French language is not so high because the French Gigaword corpus is smaller (925M words) than the English Gigaword corpus (more than 4.2G words). Consequently, a higher number of words of the text vocabulary are not in the dictionary of French word embeddings.

In order to evaluate the quality of the summaries, we use the ROUGE system[2]. More specifically, we used ROUGE-1 and ROUGE-2.

We evaluate the quality systems using DSV, CSV and our approach, which results in three versions for each system. The default version uses the cosine similarity as

[1]Site: `https://code.google.com/archive/p/word2vec/`.

[2]The options for running ROUGE 1.5.5 are -a -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0.

Datasets	compression ratio		
	n=1	n=2	n=3
English	11.7%	20.1%	25.3%
French	7.1%	12.3%	16.1%

Table A.1: *Compression ratio of vocabulary for different numbers of nearest words (n) considered with CSVs.*

similarity measure for the base, MMR and SASI systems with DSV; the TextRank system calculates the similarity between two sentences based on the content overlap of DSV. In the "cs" version, all systems use the sentence embedding representation for the sentences as described in (Kågebäck et al., 2014) and employ the cosine similarity as similarity measure. Finally, the "rv" version (our method) uses a reduced vocabulary and the same metrics as the default version with DSVs. After selecting the best sentences, all system versions create a summary with the original sentences.

Despite the good compression ratio with $n = 2$ or 3, the best summaries with a reduced vocabulary were obtained when taking into account only one nearest word and a window size of 6 for word2vec. Table A.2 shows the results for the English and French corpora. Almost all the "cs" systems using the continuous space and the reduced vocabulary are better than the default systems.

Systems	English		French	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
base	0.254	0.053	0.262	**0.059**
base_cs	**0.262**	**0.054**	0.261	0.057
base_rv	**0.262**	**0.054**	**0.264**	0.054
MMR	0.262	**0.058**	0.270	0.059
MMR_cs	0.260	0.053	**0.277***	**0.072***
MMR_rv	**0.265***	**0.058**	0.270	0.059
SASI	0.251	0.053	0.248	0.047
SASI_cs	0.247	**0.058**	**0.251**	0.047
SASI_rv	**0.253**	0.053	0.244	**0.050**
TextRank	0.251	0.056	0.267	0.063
TextRank_cs	**0.261**	0.056	**0.276**	**0.065**
TextRank_rv	0.260	**0.062***	0.268	0.058

Table A.2: *ROUGE F-scores for English and French summaries. The bold numbers are the best values for each group of systems in each metric. A star indicates the best system for each metric.*

For the English corpus, the "rv" versions obtain the best values, which indicates that the reduced vocabulary improves the quality of the similarity calculus and the statistical metrics. The difference in the results between English and French is related to the size of the corpus to create word embeddings. Since the French training corpus is not as big, the precision of the semantic word relationships is not accurate enough and the closest word may not be similar. Furthermore, the French word embedding

dictionary is smaller than for English. Consequently, the "rv" version sometimes does not find the true similar words in the continuous space and the reduced vocabulary may be incorrect. The "cs" version mitigates the problem with the small vocabulary because this version only analyzes the words of the text that exist in the continuous space. Thus the "cs" version produces better summaries for almost all systems.

All in all, our results show that the reduced vocabulary in a discrete space improves the performance of the state-of-the-art. The use of discrete context representation enables the utilization of factorization matrices methods (e.g. LSA) and order metrics implemented to the discrete representation. Unfortunately, continuous word representations do not have a degree of similarity correlated with a distance metric between two words vectors, i.e. two words that have similar context can have a cosine similarity of 0.7 or of 0.4. This threshold to determine if two words share the same context is different for each region in the space dimensional because of the frequency and the arrangement of these words in corpora where word embeddings were learned. Therefore, the discrete context vocabulary considers only the nearest words as similar and other words as independent words.

Glossary

AE AutoEncoder.

ASR Automatic Speech Recognition.

CBOW Continuous Bag-Of-Word.

CG Chunk-Graph.

CLTS Cross-Language Text Summarization.

CNN Convolutional Neural Network.

CNNLM Convolutional Neural Network Language Model.

CR Compression Ratio.

CSV Continuous Space Vector.

DSV Discrete Space Vector.

DT-RNN Dependency Tree Recurrent Neural Network.

FFNN FeedForward Neural Network.

GRU Gated Recurrent Unit.

ILP Integer Linear Programming.

IR Information Retrieval.

LDA Latent Dirichlet Allocation.

LSA Latent Semantic Analysis.

LSTM Long Short Term Memory.

MDS Multi-Document Summarization.

MMR Maximal Marginal Relevance.

MSC Multi-Sentence Compression.

MSE Mean Squared Error.

MT Machine Translation.

MWE Multi-Word Expression.

NLP Natural Language Processing.

NMT Neural Machine Translation.

NN Neural Network.

POS Part-of-Speech.

RAE Recursive AutoEncoder.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

RNNLM Recurrent Neural Network Language Model.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

SBD Sentence Boundary Detection.

SC Sentence Compression.

SDT-RNN Semantic Dependency Tree Recurrent Neural Network.

STS Semantic Text Similarity.

SVM Support Vector Machine.

SVR Support Vector Regression.

TF-IDF Term Frequency-Inverse Document Frequency.

TF-ISF Term Frequency-Inverse Sentence Frequency.

TS Text Summarization.

TTR Type-Token Ratio.

WER Word Error Rate.

WG Word Graph.

List of Figures

List of Tables

Bibliography

Agirre, E., D. Cer, M. Diab, A. Gonzalez-Agirre, & W. Guo (2013). SEM 2013 Shared Task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pp. 32–43. Association for Computational Linguistics.

Albelwi, S. & A. Mahmood (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. *Entropy 19*(6).

Almeida, M. & A. Martins (2013). Fast and Robust Compressive Summarization with Dual Decomposition and Multi-Task Learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria, pp. 196–206. Association for Computational Linguistics.

Bahdanau, D., K. Cho, & Y. Bengio (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR abs/1409.0473*.

Balikas, G. & M.-R. Amini (2015). Learning Language-Independent Sentence Representations for Multi-Lingual, Multi-Document Summarization. In *17ème Conférence Francophone sur l'Apprentissage Automatique (CAp)*.

Banerjee, S., P. Mitra, & K. Sugiyama (2015). Multi-Document Abstractive Summarization Using ILP Based Multi-Sentence Compression. In *IJCAI*, pp. 1208–1214.

Baralis, E., L. Cagliero, N. A. Mahoto, & A. Fiori (2013). GraphSum: Discovering Correlations Among Multiple Terms for Graph-based Summarization. *Inf. Sci. 249*, 96–109.

Barzilay, R. & M. Lapata (2006). Aggregation via Set Partitioning for Natural Language Generation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, Stroudsburg, PA, USA, pp. 359–366. Association for Computational Linguistics.

Barzilay, R. & K. R. McKeown (2005). Sentence Fusion for Multidocument News Summarization. *Comput. Linguist. 31*(3), 297–328.

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.

Bengio, Y., R. Ducharme, P. Vincent, & C. Janvin (2003). A Neural Probabilistic Language Model. *J. Mach. Learn. Res. 3*, 1137–1155.

Berg-Kirkpatrick, T., D. Gillick, & D. Klein (2011). Jointly Learning to Extract and Compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, Stroudsburg, PA, USA, pp. 481–490. Association for Computational Linguistics.

Bing, L., P. Li, Y. Liao, W. Lam, W. Guo, & R. J. Passonneau (2015). Abstractive Multi-Document Summarization via Phrase Selection and Merging. In *ACL (1)*, pp. 1587–1597. The Association for Computer Linguistics.

Bjerva, J., J. Bos, R. van der Goot, & M. Nissim (2014). The Meaning Factory: Formal Semantics for Recognizing Textual Entailment and Determining Semantic Similarity. In *SemEval@COLING*, pp. 642–646. The Association for Computer Linguistics.

Blei, D. M., A. Y. Ng, & M. I. Jordan (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res. 3*, 993–1022.

Bojanowski, P., E. Grave, A. Joulin, & T. Mikolov (2017a). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

Bojanowski, P., E. Grave, A. Joulin, & T. Mikolov (2017b). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

Bojanowski, P., E. Grave, A. Joulin, & T. Mikolov (2017c). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, & M. Turchi (2017). Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark, pp. 169–214. Association for Computational Linguistics.

Botha, J. A. & P. Blunsom (2014). Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1899–II–1907. JMLR.org.

Boudin, F., S. Huet, & J. Torres-Moreno (2011). A Graph-based Approach to Cross-Language Multi-Document Summarization. *Polibits 43*, 113–118.

Boudin, F. & E. Morin (2013). Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression. In *NAACL*, pp. 298–305.

Boudin, F. & J. M. Torres Moreno (2007). NEO-CORTEX: A Performant User-Oriented Multi-Document Summarization System. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing*, Berlin, Heidelberg, pp. 551–562. Springer Berlin Heidelberg.

Brin, S. & L. Page (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst. 30*(1-7), 107–117.

Brown, P., J. Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, & P. Roossin (1988). A Statistical Approach to Language Translation. In *Proceedings of the 12th Conference on Computational Linguistics - Volume 1*, COLING '88, Stroudsburg, PA, USA, pp. 71–76. Association for Computational Linguistics.

Brown, P. F., V. J. D. Pietra, S. A. D. Pietra, & R. L. Mercer (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Comput. Linguist. 19*(2), 263–311.

Bruckner, S., F. Hüffner, C. Komusiewicz, & R. Niedermeier (2013). *Evaluation of ILP-Based Approaches for Partitioning into Colorful Components*, pp. 176–187. Berlin, Heidelberg: Springer Berlin Heidelberg.

Cao, Z., F. Wei, L. Dong, S. Li, & M. Zhou (2015). Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2153–2159. AAAI Press.

Carbonell, J. & J. Goldstein (1998). The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *SIGIR*, pp. 335–336.

Chen, X., L. Xu, Z. Liu, M. Sun, & H. Luan (2015). Joint Learning of Character and Word Embeddings. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 1236–1242. AAAI Press.

Cheng, J. & M. Lapata (2016). Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Cho, K., B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, & Y. Bengio (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1724–1734. Association for Computational Linguistics.

Chopra, S., M. Auli, & A. M. Rush (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 93–98.

Christensen, H., Y. Gotoh, B. Kolluru, & S. Renals (2003). Are Extractive Text Summarisation Techniques Portable to Broadcast News? In *IEEE Workshop on Automatic Speech Recognition and Understanding ASRU'03*, pp. 489–494. IEEE.

Chung, J., C. Gulcehre, K. Cho, & Y. Bengio (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Clarke, J. & M. Lapata (2007). Modelling Compression with Discourse Constraints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and on Computational Natural Language Learning (EMNLP-CoNLL-2007)*, Prague, Czech Republic, pp. 1–11.

Cleveland, D. B. & A. D. Cleveland (1983). *Introduction to Indexing and Abstracting*. Littleton, Colo. : Libraries Unlimited. Includes index.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, & P. Kuksa (2011). Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res. 12*, 2493–2537.

Dasgupta, A., R. Kumar, & S. Ravi (2013). Summarization Through Submodularity and Dispersion. In *ACL (1)*, pp. 1014–1022. The Association for Computer Linguistics.

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.

Edmundson, H. P. (1969). New Methods in Automatic Extracting. *J. ACM 16*(2), 264–285.

Erkan, G. & D. R. Radev (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *J. Artif. Intell. Res. 22*, 457–479.

Ermakova, L. & J. Mothe (2016). Query Expansion by Local Context Analysis. In *CORIA 2016 - Conférence en Recherche d'Informations et Applications- 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, Toulouse, France, March 9-11, 2016.*, pp. 235–250.

Fang, C., D. Mu, Z. Deng, & Z. Wu (2017). Word-sentence Co-ranking for Automatic Extractive Text Summarization. *Expert Syst. Appl. 72*(C), 189–195.

Ferreira, R., L. de Souza Cabral, F. L. G. de Freitas, R. D. Lins, G. de França Pereira e Silva, S. J. Simske, & L. Favaro (2014). A Multi-Document Summarization System based on Statistics and Linguistic Treatment. *Expert Syst. Appl. 41*(13), 5780–5787.

Filippova, K. (2010). Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. In *COLING*, pp. 322–330.

Filippova, K., E. Alfonseca, C. A. Colmenares, L. Kaiser, & O. Vinyals (2015). Sentence Compression by Deletion with LSTMs. In *EMNLP*, pp. 360–368.

Filippova, K. & M. Strube (2008). Sentence Fusion via Dependency Graph Compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Stroudsburg, PA, USA, pp. 177–185. Association for Computational Linguistics.

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*, Marrakech, Morocco.

Furui, S., T. Kikuchi, Y. Shinnaka, & C. Hori (2004). Speech-to-Text and Speech-to-Speech Summarization of Spontaneous Speech. *IEEE Transactions on Speech and Audio Processing 12*(4), 401–408.

Galanis, D., G. Lampouras, & I. Androutsopoulos (2012). Extractive Multi-Document Summarization with Integer Linear Programming and Support Vector Regression. In *COLING*.

Galley, M. & K. McKeown (2007). Lexicalized Markov Grammars for Sentence Compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 180–187. Association for Computational Linguistics.

Garey, M. R. & D. S. Johnson (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Genest, P.-E. & G. Lapalme (2012). Fully Abstractive Approach to Guided Summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, Stroudsburg, PA, USA, pp. 354–358. Association for Computational Linguistics.

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

Gillick, D. & B. Favre (2009). A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, ILP '09, Stroudsburg, PA, USA, pp. 10–18. Association for Computational Linguistics.

Gong, Y. & X. Liu (2001). Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, New York, NY, USA, pp. 19–25. ACM.

González-Gallardo, C. & J. Torres-Moreno (2018). Sentence Boundary Detection for French with Subword-Level Information Vectors and Convolutional Neural Networks. *CoRR abs/1802.04559*.

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

Greff, K., R. K. Srivastava, J. Koutník, B. R. Steunebrink, & J. Schmidhuber (2015). LSTM: A Search Space Odyssey. *CoRR abs/1503.04069*.

Gülçehre, Ç., S. Ahn, R. Nallapati, B. Zhou, & Y. Bengio (2016). Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Hahnloser, R. H. R., R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, & H. S. Seung (2000). Digital Selection and Analogue Amplification Coexist in a Cortex-Inspired Silicon Circuit. *Nature 405*, 947–951.

Hanson, S. J. & D. J. Burr (1990). What Connectionist Models Learn: Learning and Representation in Connectionist Networks. *Behavioral and Brain Sciences 13*(3), 471–489.

He, H., K. Gimpel, & J. J. Lin (2015). Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1576–1586.

Hertz, J., R. G. Palmer, & A. S. Krogh (1991). *Introduction to the Theory of Neural Computation* (1st ed.). Perseus Publishing.

Hochreiter, S. & J. Schmidhuber (1997). Long Short-Term Memory. *Neural Comput. 9*(8), 1735–1780.

Hu, B., Q. Chen, & F. Zhu (2015). LCSTS: A Large Scale Chinese Short Text Summarization Dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1967–1972.

Huet, S., G. Gravier, & P. Sébillot (2010). Morpho-Syntactic Post-Processing of N-Best Lists for Improved French Automatic Speech Recognition. *Computer Speech and Language 24*(4), 663–684.

Indurkhya, N. & F. J. Damerau (2010). *Handbook of Natural Language Processing* (2nd ed.). Chapman & Hall/CRC.

Jones, G. J. F., S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, & N. Ferro (Eds.) (2017). *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings*, Volume 10456 of *Lecture Notes in Computer Science*. Springer.

Jorge, C., M. L. del Rosario, & T. A. S. Pardo (2010). Experiments with CST-based Multidocument Summarization. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-5, Stroudsburg, PA, USA, pp. 74–82. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751.

Kiros, R., Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, & S. Fidler (2015). Skip-thought Vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, Cambridge, MA, USA, pp. 3294–3302. MIT Press.

Knight, K. & D. Marcu (2002). Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artif. Intell. 139*(1), 91–107.

Koehn, P. (2010). *Statistical Machine Translation* (1st ed.). New York, NY, USA: Cambridge University Press.

Koehn, P., F. J. Och, & D. Marcu (2003). Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, Stroudsburg, PA, USA, pp. 48–54. Association for Computational Linguistics.

Kovář, V., A. Horák, & M. Jakubíček (2009). Syntactic Analysis using Finite Patterns: A New Parsing System for Czech. In *Language and Technology Conference*, pp. 161–171. Springer.

Kulkarni, N. & M. A. Finlayson (2011). jMWE: a Java toolkit for Detecting Multi-Word Expressions. In *Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE)*, pp. 122–124.

Kågebäck, M., O. Mogren, N. Tahmasebi, & D. Dubhashi (2014). Extractive Summarization using Continuous Vector Space Models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) EACL, April 26-30, 2014 Gothenburg, Sweden*, pp. 31–39.

Lai, A. & J. Hockenmaier (2014). Illinois-LH: A Denotational and Distributional Approach to Semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pp. 329–334.

Le, Q. & T. Mikolov (2014). Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1188–II–1196. JMLR.org.

LeCun, Y. & Y. Bengio (1998). The Handbook of Brain Theory and Neural Networks. Chapter Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. Cambridge, MA, USA: MIT Press.

Leuski, A., C.-Y. Lin, L. Zhou, U. Germann, F. J. Och, & E. Hovy (2003). Cross-lingual C*ST*RD: English Access to Hindi Information. *2*(3), 245–269.

Li, B., A. Drozd, T. Liu, & X. Du (2018). Subword-Level Composition Functions for Learning Word Embeddings. In *Proceedings of The 2nd Workshop on Subword and Character level models in NLP (SCLeM)*, pp. 38–48. ACL.

Li, C., F. Liu, F. Weng, & Y. Liu (2013). Document Summarization via Guided Sentence Compression. In *EMNLP*.

Li, C., Y. Liu, F. Liu, L. Zhao, & F. Weng (2014). Improving Multi-documents Summarization by Sentence Compression based on Expanded Constituent Parse Trees. In *EMNLP*, pp. 691–701. ACL.

Li, P., W. Lam, L. Bing, & Z. Wang (2017). Deep Recurrent Generative Decoder for Abstractive Text Summarization. In *EMNLP*, pp. 2091–2100. Association for Computational Linguistics.

Li, Q. & J. Racine (2003). Nonparametric Estimation of Distributions with Categorical and Continuous Data. *Journal of Multivariate Analysis 86*(2), 266–292.

Lin, C.-Y. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pp. 74–81.

Lin, H. & J. Bilmes (2011). A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, Stroudsburg, PA, USA, pp. 510–520. Association for Computational Linguistics.

Ling, W., I. Trancoso, C. Dyer, & A. W. Black (2015). Character-based Neural Machine Translation. *CoRR abs/1511.04586*.

Liou, C.-Y., W.-C. Cheng, J.-W. Liou, & D.-R. Liou (2014). Autoencoder for Words. *Neurocomputing 139*, 84 – 96.

Liu, L., Y. Lu, M. Yang, Q. Qu, J. Zhu, & H. Li (2018). Generative Adversarial Network for Abstractive Text Summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.

Lovász, L. (1983). *Submodular Functions and Convexity*, pp. 235–257. Berlin, Heidelberg: Springer Berlin Heidelberg.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM J. Res. Dev. 2*(2), 159–165.

Luong, A., N. Tran, V. Ung, & M. Nghiem (2015b). Word Graph-Based Multi-Sentence Compression: Re-ranking Candidates Using Frequent Words. In B. Mérialdo, M. L. Nguyen, D. Le, D. A. Duong, & S. Tojo (Eds.), *2015 Seventh International Conference on Knowledge and Systems Engineering, KSE 2015, Ho Chi Minh City, Vietnam, October 8-10, 2015*, pp. 55–60. IEEE.

Luong, T., H. Pham, & C. D. Manning (2015a). Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods*

in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pp. 1412–1421.

Manning, C., M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, & D. McClosky (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pp. 55–60.

Marelli, M., S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, & R. Zamparelli (2014). A SICK Cure for the Evaluation of Compositional Distributional Semantic Models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pp. 216–223.

Martins, A. F. T. & N. A. Smith (2009). Summarization with a Joint Model for Sentence Extraction and Compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, Stroudsburg, PA, USA, pp. 1–9. Association for Computational Linguistics.

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.

McDonald, R. (2007). A Study of Global Inference Algorithms in Multi-document Summarization. In G. Amati, C. Carpineto, & G. Romano (Eds.), *Advances in Information Retrieval*, Berlin, Heidelberg, pp. 557–564. Springer Berlin Heidelberg.

McDonald, R. T. (2006). Discriminative Sentence Compression with Soft Syntactic Evidence. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*.

McKeown, K., J. Hirschberg, M. Galley, & S. Maskey (2005). From Text to Speech Summarization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*, Volume 5, pp. v–997.

McKeown, K., S. Rosenthal, K. Thadani, & C. Moore (2010). Time-efficient Creation of an Accurate Sentence Fusion Corpus. In *HLT-NAACL*, pp. 317–320.

Miao, Y. & P. Blunsom (2016). Language as a Latent Variable: Discrete Generative Models for Sentence Compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 319–328. Association for Computational Linguistics.

Mihalcea, R. & P. Tarau (2004). TextRank: Bringing Order into Texts. In D. Lin & D. Wu (Eds.), *Proceedings of EMNLP 2004*, Barcelona, Spain, pp. 404–411. Association for Computational Linguistics.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, & J. Dean (2013). Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, USA, pp. 3111–3119. Curran Associates Inc.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *COMMUNICATIONS OF THE ACM 38*, 39–41.

Moens, M.-F., R. Angheluta, R. Mitra, & X. Jing (2004). K.U.Leuven Summarization System at DUC 2004. Document Understanding Conference, Boston, 2004.

Moirón, B. V. & J. Tiedemann (2006). Identifying Idiomatic Expressions using Automatic Word-Alignment. In *EACL 2006 Workshop on Multiword Expressions in a Multilingual Context*.

Mou, L., Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, & Z. Jin (2016). How Transferable are Neural Networks in NLP Applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 479–489. Association for Computational Linguistics.

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.

Mueller, J. & A. Thyagarajan (2016). Siamese Recurrent Architectures for Learning Sentence Similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2786–2792. AAAI Press.

Nair, V. & G. E. Hinton (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, USA, pp. 807–814. Omnipress.

Nallapati, R., F. Zhai, & B. Zhou (2017). SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 3075–3081.

Nallapati, R., B. Zhou, C. N. dos Santos, Ç. Gülçehre, & B. Xiang (2016). Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *CoNLL*, pp. 280–290. ACL.

Napoles, C., B. Van Durme, & C. Callison-Burch (2011). Evaluating Sentence Compression: Pitfalls and Suggested Remedies. In *Workshop on Monolingual Text-To-Text Generation (MTTG)*, pp. 91–97.

Nayeem, M. T., T. A. Fuad, & Y. Chali (2018). Abstractive Unsupervised Multi-Document Summarization using Paraphrastic Sentence Fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1191–1204. Association for Computational Linguistics.

Nenkova, A., R. Passonneau, & K. McKeown (2007). The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. *ACM Trans. Speech Lang. Process. 4*(2).

Niu, J., H. Chen, Q. Zhao, L. Su, & M. Atiquzzaman (2017). Multi-Document Abstractive Summarization using Chunk-Graph and Recurrent Neural Network. In *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6.

Och, F. J. & H. Ney (2004). The Alignment Template Approach to Statistical Machine Translation. *Comput. Linguist. 30*(4), 417–449.

Och, F. J. & H. Weber (1998). Improving Statistical Natural Language Translation with Categories and Rules. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, Stroudsburg, PA, USA, pp. 985–989. Association for Computational Linguistics.

Oliveira, H., R. D. Lins, R. Lima, & F. Freitas (2017). A Regression-Based Approach Using Integer Linear Programming for Single-Document Summarization. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 270–277.

Öncan, T., İ. K. Altınel, & G. Laporte (2009). A Comparative Analysis of Several Asymmetric Traveling Salesman Problem Formulations. *Computers & Operations Research 36*(3), 637–654.

Orasan, C. & O. A. Chiorean (2008). Evaluation of a Cross-lingual Romanian-English Multi-document Summariser. In *6th International Conference on Language Resources and Evaluation (LREC)*.

Paulus, R., C. Xiong, & R. Socher (2017). A Deep Reinforced Model for Abstractive Summarization. *CoRR abs/1705.04304*.

Pavlick, E., P. Rastogi, J. Ganitkevitch, B. Van Durme, & C. Callison-Burch (2015). PPDB 2.0: Better Paraphrase Ranking, Fine-grained Entailment Relations, Word Embeddings, and Style Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 425–430. Association for Computational Linguistics.

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.

Phung, V. & L. De Vine (2015). A Study on the Use of Word Embeddings and PageRank for Vietnamese Text Summarization. In *20th Australasian Document Computing Symposium*, pp. 7:1–7:8.

Qian, X. & Y. Liu (2013). Fast Joint Compression and Summarization via Graph Cuts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1492–1502. Association for Computational Linguistics.

Reape, M. & C. Mellish (1999). Just What is Aggregation Anyway? In P. S. Dizier (Ed.), *7th European Workshop on Natural Language Generation*, Toulouse, pp. 20–29.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptions and the Theory of Brain Mechanisms*. Washington, DC: Spartan.

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, A. M., S. Chopra, & J. Weston (2015). A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 379–389.

Rychalska, B., K. Pakulska, K. Chodorowska, W. Walczak, & P. Andruszkiewicz (2016). Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity. In *SemEval@ NAACL-HLT*.

Sadat, F., A. Maeda, M. Yoshikawa, & S. Uemura (2002). A Combined Statistical Query Term Disambiguation in Cross-Language Information Retrieval. In *Proceedings. 13th International Workshop on Database and Expert Systems Applications*, pp. 251–255.

Saggion, H. & G. Lapalme (2002). Generating Indicative-informative Summaries with sumUM. *Comput. Linguist. 28*(4), 497–526.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. McGraw-Hill, New York.

Schmid, H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT Workshop*, pp. 47–50.

See, A., P. J. Liu, & C. D. Manning (2017). Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083. Association for Computational Linguistics.

Sennrich, R., B. Haddow, & A. Birch (2016). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Severyn, A., M. Nicosia, & A. Moschitti (2013). Learning Semantic Textual Similarity with Structural Representations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 714–718. Association for Computational Linguistics.

ShafieiBavani, E., M. Ebrahimi, R. K. Wong, & F. Chen (2016). An Efficient Approach for Multi-Sentence Compression. In R. J. Durrant & K.-E. Kim (Eds.), *Proceedings of The 8th Asian Conference on Machine Learning*, Volume 63 of *Proceedings of Machine Learning Research*, The University of Waikato, Hamilton, New Zealand, pp. 414–429. PMLR.

Shanmugamani, R. (2018). *Deep Learning for Computer Vision*. Packt Publishing.

Sipos, R., P. Shivaswamy, & T. Joachims (2012). Large-margin Learning of Submodular Summarization Models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, Stroudsburg, PA, USA, pp. 224–233. Association for Computational Linguistics.

Socher, R., A. Karpathy, Q. V. Le, C. D. Manning, & A. Y. Ng (2014). Grounded Compositional Semantics for Finding and Describing Images with Sentences. *TACL 2*, 207–218.

Spärck-Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation 1*(28), 11–21.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, & R. Salakhutdinov (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res. 15*(1), 1929–1958.

Sun, R., Y. Zhang, M. Zhang, & D.-H. Ji (2015). Event-Driven Headline Generation. In *ACL*.

Tai, K. S., R. Socher, & C. D. Manning (2015). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *CoRR abs/1503.00075*, 1556–1566.

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.

Thadani, K. & K. McKeown (2013). Supervised Sentence Fusion with Single-Stage Inference. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pp. 1410–1418.

Torres-Moreno, J.-M. (2014). *Automatic Text Summarization*. London: Wiley and 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 *Proceedings of the Seventh Symposium on Information and Communication Technology*, SoICT '16, New York, NY, USA, pp. 123–130. ACM.

Tsubaki, M., K. Duh, M. Shimbo, & Y. Matsumoto (2016). Non-Linear Similarity Learning for Compositionality. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pp. 2828–2834.

Turian, J., L. Ratinov, & Y. Bengio (2010). Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, Stroudsburg, PA, USA, pp. 384–394. Association for Computational Linguistics.

Tzouridis, E., J. A. Nasir, & U. Brefeld (2014). Learning to Summarise Related Sentences. In *COLING*, pp. 1636–1647. ACL.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, & I. Polosukhin (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc.

Wan, X. (2011). Using Bilingual Information for Cross-Language Document Summarization. In *ACL*, pp. 1546–1555. The Association for Computer Linguistics.

Wan, X., H. Li, & J. Xiao (2010). Cross-Language Document Summarization Based on Machine Translation Quality Prediction. In *ACL*.

Wan, X., F. Luo, X. Sun, S. Huang, & J.-g. Yao (2018). Cross-Language Document Summarization via Extraction and Ranking of Multiple Summaries. *Knowledge and Information Systems*.

Wang, L., H. Raghavan, V. Castelli, R. Florian, & C. Cardie (2013). A Sentence Compression Based Framework to Query-Focused Multi-Document Summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria, pp. 1384–1394. Association for Computational Linguistics.

Wang, L., J. Yao, Y. Tao, L. Zhong, W. Liu, & Q. Du (2018). A Reinforced Topic-Aware Convolutional Sequence-to-Sequence Model for Abstractive Text Summarization. *CoRR abs/1805.03616*, 4453–4460.

Woodsend, K. & M. Lapata (2012). Multiple Aspect Summarization Using Integer Linear Programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 233–243. Association for Computational Linguistics.

Xiong, W., J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, & G. Zweig (2016). Achieving Human Parity in Conversational Speech Recognition. *PP*.

Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, & Y. Bengio (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, Volume 37 of *Proceedings of Machine Learning Research*, Lille, France, pp. 2048–2057. PMLR.

Xu, Y.-D., X.-D. Zhang, G.-R. Quan, & Y. Wang (2013). MRS for Multi-Document Summarization by Sentence Extraction. *Telecommunication Systems 53*, 91–98.

Yao, J., X. Wan, & J. Xiao (2015a). Compressive Document Summarization via Sparse Optimization. In *IJCAI*, pp. 1376–1382. AAAI Press.

Yao, J., X. Wan, & J. Xiao (2015b). Phrase-based Compressive Cross-Language Summarization. In *EMNLP*, pp. 118–127.

Yin, W. & Y. Pei (2015). Optimizing Sentence Modeling and Selection for Document Summarization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 1383–1389. AAAI Press.

Yousefi-Azar, M. & L. Hamey (2017). Text Summarization using Unsupervised Deep Learning. *Expert Syst. Appl. 68*(C), 93–105.

Yuan, Z. & T. Briscoe (2016). Grammatical Error Correction using Neural Machine Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 380–386. Association for Computational Linguistics.

Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *CoRR abs/1212.5701*.

Zhang, J., Y. Zhou, & C. Zong (2016). Abstractive Cross-Language Summarization via Translation Model Enhanced Predicate Argument Structure Fusing. *IEEE/ACM Trans. Audio, Speech & Language Processing 24*(10), 1842–1853.

Zhao, J., T. Zhu, & M. Lan (2014). ECNU: One Stone Two Birds: Ensemble of Heterogenous Measures for Semantic Relatedness and Textual Entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 271–277. Association for Computational Linguistics.

Zheng, C., K. Swenson, E. Lyons, & D. Sankoff (2011). OMG! Orthologs in Multiple Genomes — Competing Graph-Theoretical Formulations. In T. M. Przytycka & M.-F. Sagot (Eds.), *WABI*, Berlin, Heidelberg, pp. 364–375. Springer Berlin Heidelberg.

Zhu, Q., X. Li, A. Conesa, & C. Pereira (2018). GRAM-CNN: a Deep Learning Approach with Local Context for Named Entity Recognition in Biomedical Text. *Bioinformatics 34*(9), 1547–1554.

Personal Bibliography

González-Gallardo, C.-E., E. Linhares Pontes, F. Sadat, & J.-M. Torres-Moreno (2018). Automated Sentence Boundary Detection in Modern Standard Arabic Transcripts using Deep Neural Networks. In *4th International Conference on Arabic Computational Linguistics (ACLing 2018)- (Submission)*.

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.

Linhares Pontes, E., C.-E. González-Gallardo, J.-M. Torres-Moreno, & S. Huet (2019). Cross-Lingual Speech-to-Text Summarization. In K. Choroś, M. Kopel, E. Kukla, & A. Siemiński (Eds.), *Multimedia and Network Information Systems*, Cham, pp. 385–395. Springer International Publishing.

Linhares Pontes, E., T. Gouveia da Silva, A. C. Linhares, J.-M. Torres-Moreno, & S. Huet (2016). Métodos de Otimização Combinatória Aplicados ao Problema de Compressão MultiFrases. In *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, pp. 2278–2289.

Linhares Pontes, E., S. Huet, T. Gouveia da Silva, A. C. Linhares, & J.-M. Torres-Moreno (2018c). Multi-Sentence Compression with Word Vertex-Labeled Graphs and Integer Linear Programming. In *Proceedings of TextGraphs-12: the Workshop on Graph-based Methods for Natural Language Processing*. Association for Computational Linguistics.

Linhares Pontes, E., S. Huet, A. C. Linhares, & J.-M. Torres-Moreno (2018d). Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. In *25e Conférence sur le Traitement Automatique des Langues Naturelles*.

Linhares Pontes, E., S. Huet, & J.-M. Torres-Moreno (2018a). Microblog Contextualization: Advantages and Limitations of a Multi-sentence Compression Approach. In P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J. Y. Nie, L. Soulier, E. SanJuan, L. Cappellato, & N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Cham, pp. 181–190. Springer International Publishing.

Linhares Pontes, E., S. Huet, & J.-M. Torres-Moreno (2018b). A Multilingual Study of Compressive Cross-Language Text Summarization. In *17th Mexican International Conference on Artificial Intelligence (MICAI)*, Guadalajara, Mexico. Springer.

Linhares Pontes, E., S. Huet, J.-M. Torres-Moreno, & A. C. Linhares (2016). Automatic Text Summarization with a Reduced Vocabulary Using Continuous Space Vectors. In *NLDB*, Volume 9612 of *Lecture Notes in Computer Science*, pp. 440–446. Springer.

Linhares Pontes, E., S. Huet, J.-M. Torres-Moreno, & A. C. Linhares (2017). Microblog Contextualization using Continuous Space Vectors: Multi-Sentence Compression of Cultural Documents. In *Working Notes of the CLEF Lab on Microblog Cultural Contextualization*, Volume 1866. CEUR-WS.org.

Linhares Pontes, E., A. C. Linhares, & J.-M. Torres-Moreno (2014). SASI: Sumarizador Automático de Documentos Baseado no Problema do Subconjunto Independente de Vértices. In *XLVI Simpósio Brasileiro de Pesquisa Operacional*.

Linhares Pontes, E., J.-M. Torres-Moreno, S. Huet, & A. C. Linhares (2016). Tweet Contextualization using Continuous Space Vectors: Automatic Summarization of Cultural Documents. In *Working Notes of the CLEF Lab on Microblog Cultural Contextualization*, Volume 1609. CEUR-WS.org.

Linhares Pontes, E., J.-M. Torres-Moreno, S. Huet, & A. C. Linhares (2018). A New Annotated Portuguese/Spanish Corpus for the Multi-Sentence Compression Task. In N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, & T. Tokunaga (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Linhares Pontes, E., J.-M. Torres-Moreno, & A. C. Linhares (2015). LIA-RAG: a System Based on Graphs and Divergence of Probabilities Applied to Speech-to-Text Summarization. In M. . P. Addendum (Ed.), *Multiling CCCS*.

Ménard, E., N. Khashman, S. Kochkina, J.-M. Torres-Moreno, P. Velazquez-Morales, F. Zhou, P. Jourlin, P. Rawat, P. Peinl, E. Linhares Pontes, & I. Brunetti (2016). A Second Life for TIIARA: From Bilingual to Multilingual! *Knowledge Organization 43*(1), 22–34.

Smaïli, K., D. Fohr, C.-E. González-Gallardo, M. Grega, L. Janowski, D. Jouvet, A. Komorowski, A. Koźbiał, D. Langlois, M. Leszczuk, O. Mella, M. A. Menacer, A. Mendez, E. Linhares Pontes, E. SanJuan, D. Świst, J.-M. Torres-Moreno, & B. Garcia-Zapirain (2019). A First Summarization System of a Video in a Target Language. In K. Choroś, M. Kopel, E. Kukla, & A. Siemiński (Eds.), *Multimedia and Network Information Systems*, Cham, pp. 77–88. Springer International Publishing.