POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Timeline summarization in a cross-lingual scenario

Supervisors

Prof. Luca CAGLIERO

Prof. Moreno LA QUATRA

Candidate

Jana MAGDESKA

April 2021

Abstract

Timeline Summarization (TLS) is the main approach that has been used to automatically create timelines of news reports related to some long-running event of interest. These timelines present a good way to keep up-to-date to the topic and follow its development over time, while getting only the salient information related to it. Most of the previous work done on TLS is focused on creating timelines for resources in a monolingual scenario, typically in the English language, and although some of them can be portable to other languages, they are still unable to combine the knowledge that has been extracted from news articles in different languages. The main contribution of this Thesis work is the study and development of TLS strategies tailored to multilingual resources. It aims at combining the textual information from different languages in order to generate a news timeline in a given target language. Among the biggest challenges addressed in this study is the analysis of these multilingual resources, as well as exploring the possibility to use the additional knowledge provided among the languages. The motivation for this is based on the fact that news provided in different languages can demonstrate a different aspect of the same event, as they may portray different cultural, economical or political reflections. The use of this additional knowledge gained from the multilingual resources has been proved to be beneficial, enriching the resources of the various languages and increasing their summarization quality significantly.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Luca Cagliero, for guiding me through the process of working on this thesis, giving me valuable suggestions and ideas.
I would also like to thank my co-supervisor, Moreno La Quatra, for his support and assistance during my work, and for always finding time to clarify any doubt I had.

I am deeply grateful to my family, my parents and my sister, for their understanding and continuous encouragement throughout my studies. I would also like to extend my deepest gratitude to Alessandro, who has patiently supported me during this entire journey.

Finally, I would like to thank my friends who have made the study breaks something to look forward to, and this whole process less stressful.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction to summarization

Reading news reports that cover some event of interest could be a good way to stay informed on the topic and follow how the event develops over time. Most of the published news articles contain a timestamp which can be used to keep the chronology of the event. However, there is a large amount of news articles related to an event, published (almost) on a daily basis by many news agencies, especially in the case of long-running events. As readers, we are often interested only in the most important facts related to the event, and going through all the news reports could be rather exhaustive.

The process of examining a large set of documents (in our case news articles), in order to extract new information from it, is known as *text mining*. This process can identify the facts or relationships from the text, and convert it into a structured form for further analysis. Among the different methodologies that could be applied to process the text, is the *Natural Language Processing (NLP)*, which is in charge of helping the computers understand, interpret and perform various manipulations of the human language.

In order to have a shorter version of the news articles, while simultaneously keeping the most important points and the meaning of the content, the task of *text summarization* can be applied. Given the fact that performing this task manually is quite time consuming and requires a lot of effort, the *automatic text summarization* is the most common approach for this task.

1.2 Timeline Summarization

In the context of news events, we are interested in constructing a dated summary for the most important dates that are related to a specific event. The process which automatically creates a timeline of an event that ran over a long period of time, is known as *Timeline Summarization* (TLS).

An important aspect of the TLS process, which differentiates it from the multi-document text summarization processes, is the temporal characteristic of the task. Namely, the important dates need to be identified from the collection of timestamped news articles, and a daily summary should be generated for each of them.

Fundamentally, there are two main tasks which are addressed by the TLS process: *date selection* and *date summarization*. The former refers to the process of selecting a subset of the most important dates, while the latter is in charge of building a summary for these selected dates.

1.3 Multilingual resources

Often, the news related to a specific event are reported in many different languages, thus providing an even larger amount of resources. Each of these news reports can demonstrate a different aspect related to the same event, as they may portray different cultural, economical or political reflections.

Having in mind that the topics of interest in this study are related to armed conflicts, it's quite common that news agencies from different countries may report the same event from a different perspective. Thus, the motivation for using multilingual resources appeared.

1.4 Objectives of the Thesis work

Most of the previous work on TLS has been based on several publicly available datasets, containing news articles in the English language. Although some of them can be portable to other languages, they are still unable to combine knowledge that has been extracted from news articles in different languages.

The main contribution of this Thesis work is the proposal of performing the TLS process using multilingual resources, by combining the textual information from different languages in order to generate a news timeline in a given target language. The goal is to enrich the target language resources, providing an additional knowledge from the other languages, and improve the quality of the summaries in general.

The additional knowledge is provided by extending the target language resources, with those of another source language, therefore resulting with cross-lingual resources. This is done by translating the resources from one or more source languages, to a target language.

1.5 Proposed methodology and results

Different methodologies have been proposed in order to reach the main objective. After comparing the summarization quality of the languages in the monolingual scenario, various translation techniques have been applied in order to enrich the language resources. Based on where the translation occurs in the pipeline of the TLS process, the main techniques can be divided into:

- *Early translation*: the source documents are translated to the target language before the date selection and summarization phases
- *Mid translation*: documents are first summarized, their summaries are translated and at the end the date selection phase is done (and a further summarization, if needed)
- *Late translation*: translation is done after the date selection and summarization phases

As it was demonstrated in the presentation of the results in this Thesis work, the use of cross-lingual resources does indeed benefit the lower-resource languages, increasing the summarization quality significantly. The *Late Translation* has shown, on average, as the best performing among the various translation techniques that have been used in the experiments.

1.6 Thesis structure

The Thesis organization and the covered topics, can be seen as follows:

- Chapter 1 offers Introduction to the main topic of this Thesis work, as well as the motivations behind it.
- Chapter 2 introduces the preliminaries on text summarization, TLS and cross-lingual TLS.
- Chapter 3 discusses the related works which address the considered topics and strategies.
- Chapter 4 provides a detailed explanation of the process of Data collection and preprocessing, as well as the construction of the dataset which will be used for the further work.
- Chapter 5 explains the proposed methodology and all of its steps.
- Chapter 6 presents the experiments that have been carried out and the obtained results for each of them.
- Chapter 7 provides a recap of the study, as well as offers the future steps on the specific topic.

Chapter 2

Preliminaries on text summarization

2.1 Multi-document text summarization

As described in the previous chapter, text summarization is the fundamental concept on which the construction of news timelines is based. Usually, as it is the case with the news articles, we are interested into summarizing information from many documents. When an event is reported on, many news agencies publish news articles related to it, therefore multiple resources are available.

The *automatic text summarization* systems, introduced in Chapter 1, based on the input size can be classified into single-document or multi-document. The input size takes into account the number of source documents that are considered as input for the summary generation. The objective of the Single-document summarization (SDS) is to generate a summary from a single document. Multi-document summarization (MDS) on the other hand, is a process that automatically extracts the essential information from multiple documents, while removing the repetitive content from them.

Based on the text summarization technique, the following approaches for MDS (and text summarization in general) can be identified: *extractive, abstractive* or *hybrid*. The extractive approach is limited only on the content already present in the input resources and chooses the sentences that best represent it. The abstractive approach is more advanced and closer to the human-like interpretation,

and the generated summary includes sentences which were not present in the input resources. Finally, the hybrid approach combines the previous two approaches. Most of the work done on multi-document text summarization is focused on using the automatic extractive text summarization.

2.2 Timeline Summarization task

Automatic text summarization (ATS) can be used for a variety of applications, such as news, email or domain-based summarization, and each of them uses a different type of text as input: news articles, books, email, reviews etc. The Timeline Summarization (TLS) is among the various applications of ATS, using news articles as input resources.

TLS is a process which automatically creates a timeline of an event that ran over a long period of time, by creating dated daily summaries for the most important dates. Fundamentally, the two main tasks which are addressed by TLS are:

- *date selection*: select a subset of the most important dates
- *date summarization*: build a daily summary for the selected dates

Regarding the *date selection* task, the available dates are ranked based on their importance, which can take several factors into account such as the number of published articles or the number of sentences referencing to that date.

TLS has several similarities to multi-document summarization (MDS), but one important characteristic that is the basic difference between them, is that it takes into account the temporal aspect of the task, because the main task here is to choose only the most important dates from the event for the creation of the summary.

The work of Ghalandari and Ifrim, 2020 [1] identifies three different strategies for the TLS process:

- *Direct summarization*: the news article collection is treated as one set of sentences from which a timeline is directly extracted (e.g. by optimization of sentence combination [2] or by sentence ranking [3]).
- *Date-wise approach*: first select the important dates and then build a summary for each of them.
- *Event detection*: first detect the events contained in the article collection,

identify the most important ones and at the end summarize them individually.

The Date-wise strategy will be the approach of interest for our experiments, and a general overview of it can be seen on Fig. 2.1. As we can see, the process contains two main tasks, the *date selection* and *date summarization* phase. The date selection phase first defines the set of dates from where the selection can be performed, which is done by extracting the dates from the news articles. After that, a ranking is done and the most important dates are selected. This set of top l ranked dates is the set of dates for which the date summarization will be performed. After having selected the top l dates, the set of candidate sentences can be obtained by collecting the sentences that are referring to each of those dates.

The date summarization starts by representing each of the candidate sentences for a specific date as a dense vector, called *sentence embedding*, so that the summarization algorithm can be employed. The candidate sentences are ranked using an algorithm of choice, and the top ranked sentences are selected until the desired summary length k is reached. This phase is repeated for each of the top l dates selected previously.

The final summary is constructed by concatenating the daily summaries for the top ranked dates, therefore it contains l daily summaries, each with a maximum length of k, resulting with a total length of $m = l * k$.

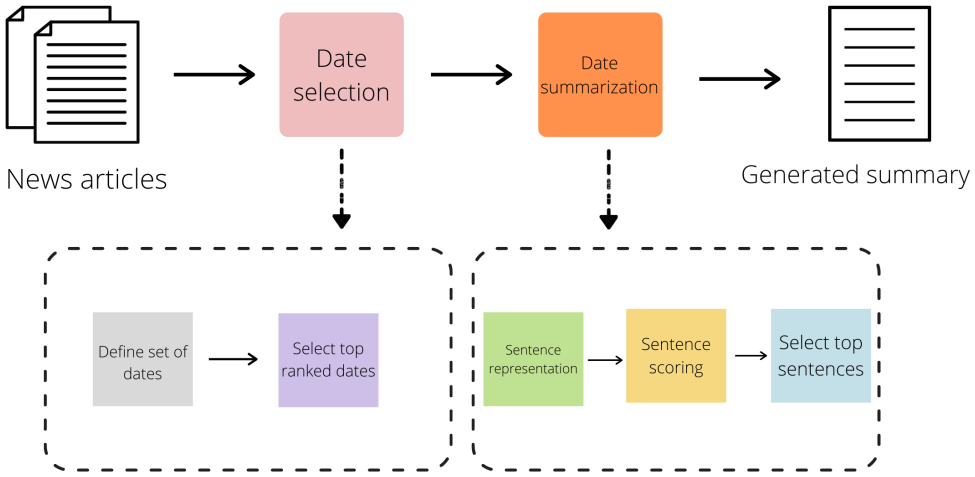

Figure 2.1: Overview of the TLS process

The Date-wise strategy, along with the more detailed explanation of the TLS process, will be covered in Chapter 5.

2.3 Cross-lingual TLS

Based on the summary language, the text summarization systems can be classified into *monolingual, multilingual*, or *cross-lingual*. In the monolingual system the language of both the source and target documents is the same, while in the multilingual one, the source documents are in several various languages and the target summaries are generated in those languages. Lastly, in the cross-lingual scenario, the source documents are in one source language and the target summary is produced in another target language.

Most of the previous work on TLS has been based on the two publicly available corpora *TL17* [4] and *crisis* [5], both containing news articles in the English language. Therefore, usually the TLS process has been considered as a monolingual problem, focusing only on one language of interest.

A different methodology has been proposed for the scope of this Thesis work, which is performing TLS with multilingual resources, using a newly proposed dataset which contains resources in three languages of interest: *Spanish, French* and *Italian*.

The cross-lingual approach is examined by combining the textual information from the different languages in order to generate a news timeline in a given target language. The motivation for this is based on the assumption that a target language resources can be enriched providing an additional knowledge from the other languages, and improve the quality of the summaries in general.

This can be done in a way that the target language resources which will be analyzed, are extended with the ones from an additional source language, by translating them to the target language. Thus, the TLS problem is now presented in a multilingual scenario, where the input resources are in more than one language. The challenges of this approach, as well as the results of the different proposed methodologies, are described in more details over the next chapters.

Chapter 3

Related work

3.1 Traditional text summarization

As mentioned in the previous section, the *extractive text summarization* is used to extract the salient information of the news articles and use it to construct the daily summaries that contain only the top ranked sentences.

This is done by choosing a subset from all the candidate sentences from the collection. In general, the majority of the summarization methods are based on performing the following tasks:

- Build a representation of the input sentences to be summarized
- Assign scores to the sentences based on the constructed representation
- Choose the top sentences for the summary and concatenate them to generate the summary

The summarization methods can be categorized into the following classes:

3.1.1 Graph-based methods

These methods are largely influenced by the *PageRank* algorithm [6] and represent the collection of documents as a connected graph, where the nodes are formed by the sentences and the edges between them represent the similarity between the pair of sentences. In order to connect two nodes, a common approach that is used is to

measure how similar the two sentences are, and connect them if that similarity is greater than a specific threshold.

Among the most common graph-based ranking models is the *TextRank* [7] model, which represents an unsupervised algorithm used for automated text summarization. Based on the *PageRank* algorithm, it finds its use in many natural language related tasks, from automated keywords extraction to extractive text summarization. This model is not domain or language dependent, and doesn't require any language specific processing.

The basic idea which the graph models are based on is the one of *voting* or *recommendation*. When one node is connected to another one, it works as it's casting a vote towards it, the higher number of votes that are cast to a node, the higher its importance is.

There are other graph-based ranking algorithms that may be used instead of *PageRank*, such as *HITS* [8] or *Positional Function* [9].
The steps for applying a graph-based model to our context will be discussed more in depth in the Graph-based models section.

3.1.2 Clustering-based methods

The clustering-based approach groups similar text units (in our case sentences) into different clusters, and each cluster consists of multiple similar text units that represent some sub-topic.
This approach is also domain and language independent, which is one of the key advantages of using it for the text summarization problems for a collection of documents.

Among the most common clustering-based methods used for text summarization are the *Centroid-based* methods. *Centroid* within a cluster is the most representative point in it. Usually, it is the mean of all the values of the points of data in the cluster. In our context, it is a pseudo-document which contains the words in the documents, that constitute this cluster that have a number of occurrences within the cluster above a defined threshold.

The centroid-based models represent the sentences as bag-of-word (BOW) vectors with TF-IDF weights and a centroid of these vectors is used in order to represent the whole document collection (Radev et al., 2004 [10]).

This approach can be easily adapted to work at summary level, instead of

sentence level, by representing a summary as the centroid of the sentence vectors and maximize the similarity between the summary centroid and the one of the document collection. A greedy algorithm is used for finding the best summary.

Another common implementation of the Centroid-based method, in the multi-documents text summarization context, is the one proposed by Ghalandari, 2017 [11] which ranks sentences based by their cosine similarity to the centroid vector of all the sentences.

3.1.3 Itemset-based methods

Another class of summarization methods are the ones based on frequent itemsets which are extracted from the document collection [12]. The sentences to be contained in the summary are selected in a way that the sentence coverage, as well as the sentence relevance score, are considered, based on some tf-idf statistics.

Unlike the other classes of summarization methods that are mostly focused on the significance of a single word within the collection, this approach is extended to the correlations of multiple words.

3.1.4 Submodular methods

Another technique for an automatic extractive summarization is proposed by Lin and Bilmes, 2011 [13] that designed a class of submodular functions for this purpose. The submodular models perform summarization by optimizing a summary in a greedy way, using submodular objective functions that represent coverage and diversity.

There are several benefits from this approach, such as that there is a greedy algorithm for monotone submodular function maximization, where it's guaranteed that the obtained summary solution is almost as good as the best possible solution.

3.1.5 Deep-learning-based methods

In the work of Kobayashi et al., 2015 [14], a summarization system has been proposed that uses a document level similarity which is based on embeddings (distributed word representations). The document is saw as a bag-of-sentences,

where each sentence is considered as a bag-of-words. The task of the system is considered as a maximization problem of a sub-modular function, which is defined by the negative sum of the distances of the nearest neighbours on the embedding distributions.

Chen and Nguyen, 2019 [15], proposed an automatic text summarization system (for single-document summarization) that uses a reinforcement learning algorithm and an RNN (recurrent neural network) sequence model, of an encoder-extractor network architecture. The selection of the important features is done by sentence-level selective encoding technique, and after that the extraction of the summary sentences is performed.

An end-to-end training model, that is based on the Deep NLP methods, has been proposed in the work of La Quatra and Cagliero, 2020 [16].
The system architecutre that has been described, *SumTO*, aims to fine-tune pre-trained embedding models, such as BERT, by exploiting the syntactic overlap between the input sentences on one hand, and the ground-truth timeline on other. In this way, these models can be tailored based on the context.

3.2 Timeline summarization methods

Among the earliest works that has been done on TLS, is the one proposed by Swan and Allan, 2000 [17]. A statistical model has been presented in this work, that can determine for an extracted feature within the text, the relative importance of it's occurrence.
The extracted features are analyzed and ranked based on the level of content they provide, and at the end they are grouped into clusters corresponding to a specific topic (significant news events which are reported in the collection of documents).

Many of the approaches that address the TLS problem only focus on generating summaries without at the same time considering the evolutionary characteristics of the news. A novel framework has been proposed by Yan et al., 2011 [18], named Evolutionary Timeline Summarization (ETS), that aims to return the evolution trajectory along a timeline. It takes a user issued query and the returned collection as input, and it automatically outputs a timeline with summaries which represent the evolutionary trajectories on specific dates.

Considering the tasks within the Timeline Summarization process, different approaches has been proposed. Based on the scenario they are addressing they can be divided accordingly:

3.2.1 Date selection

Among the common approaches for detecting the important dates in a collection of texts, is the work of Kessler et al., 2012[19]. The temporal expressions within the texts are first recognized and normalized, and after that a machine-learning technique is applied in order to extract the salient dates related to a specific topic. The main focus in this work is the extraction of the dates, and not the event they are related to. A linguistic analyzer is used to perform a deep syntactic analysis of the text and recognize the temporal expressions.

The previous approach belongs to the category of supervised machine learning approaches, that use features which are extracted from a collection of news articles. Each of the date is scored independently of the other dates in the collection.

Contrary to that, another approach has been presented by Tran et al., 2015b [20], which is much closer to the one that has been implemented in this Thesis work. Unlike the previously discussed supervised techniques, this approach takes into account the interaction between the dates in the collection, by proposing a joint graphical model. This model is a *date reference graph*, where it is represented which date is referring to which other date. On this graph a random walk model has been implemented, that includes the frequency and the temporal distance of the references, as well as the importance of the referring sentences.

3.2.2 Date summarization

Various techniques have been used for selecting the most representative sentences to be included in the daily summaries.

The proposed method by Tran et al., 2013a [4] is a supervised machine learning approach, that exploits the manually created timelines (ground-truth timeline constructed by professional journalists) in order to train a Linear Regression model to select the relevant time points and sentences to be contained in the timeline summary.

Steen and Markert, 2019 [21], suggest an unsupervised abstractive TLS system, where the date summarization is performed using a graph based merging of sentences and their compression.

Most of the TLS works are focused on extracting the relevant sentences from the full text of a news article. Tran et al. 2015a [5], on the other hand, exploits

the headlines from the online news articles for the construction of the summary, instead of using the article body.
Additional resources have been used for the construction of the summary, such as the use of social media comments (Wang et al. 2015 [22]) or the top-ranked images related to the topic (Wang et al., 2016 [23]).

3.2.3 Full TLS methods

The work of Chieu and Lee, 2004 [3] is based on a query-based event extraction system, that places summaries along a timeline. The relevant events are extracted from a collection of documents, and the sentences from this collection are ranked, taking into account the summed similarities to the other sentences in the collection.

In the work of Nguyen et al., 2014 [24], the objective is to build thematic timelines from a multi-document collection, related to a specific query. An inter-cluster ranking algorithm is presented that selects the most important related events from multiple clusters. First, a scoring model is applied to rank the sentences describing the events, and after the ranked events are being re-ranked so that the information redundancy can be reduced.

Martschat and Markert, 2018 [2] focus on experimenting if multi-document summarization (MDS) optimization models could be used to have a good performing TLS that takes into account the temporal properties. In order to accomplish that, the submodular function optimization has been adapted for the TLS task. This approach is searching for sentence combinations from the document collection towards building a timeline.

Other kind of data over which TLS can be applied is the one presented in the work of Li and Cardie, 2014 [25], where an unsupervised framework has been proposed to construct the person's life history by creating a chronological list containing the important events based on their published tweets.
A different approach that is designed to deal with dynamic stream of large-scale tweets has been proposed in the work of Wang et al., 2015 [26]. A clustering algorithm has been used in this work to cluster the tweets, after which a summarization technique is applied to generate the online summaries.

As it was presented in the previous chapter, there are several strategies to tackle the TLS process: direct summarization, date-wise approach and event detection.

The work of Martschat and Markert, 2018 represents the state-of-the-art

method for the *direct summarization* strategy, on the two commonly used datasets for TLS: crisis and T17.

On the other hand, the work of Ghalandari and Ifrim, 2020 [1] examines the following strategies for the full TLS task:

- *date-wise approach*: that first selects the important dates and after summarizes them
- *event detection approach*: that first detects events, selects them and summarizes them individually

A new method has been proposed in this work in order to improve the date summarization in the date-wise approach. This method takes advantage of the temporal expressions so that date vectors can be derived to help filter out the candidate sentences summarizing specific dates.

With the proposed modifications, the date-wise approach in this work achieves improved state-of-the-art results on all the datasets that have been tested. The event detection approach on the other hand, outperforms the state-of-the-art work [2] on one of the three datasets that have been tested.

3.3 Evaluation metrics

Evaluating the constructed summaries is an important part of the TLS process. The evaluation of machine translation can be done using *recall, precision* and the *F-measure*. This is performed by evaluating the *candidate* text which is the output of a system, and a given *reference* text.

If we want to compare X, a set of candidate items, to Y, a set of reference items, the precision and recall can be defined as follows:

$$precision(Y|X) = \frac{X \cap Y}{|Y|} \tag{3.1}$$

$$recall(Y|X) = \frac{X \cap Y}{|X|} \tag{3.2}$$

The harmonic mean between these two measures is referred to as F-measure:

$$F - measure = 2 \times \frac{precision * recall}{precision + recall} \tag{3.3}$$

The main problem is how to define a way to compute the intersection, $X \cap Y$, between a pair of texts. The earliest approaches were computing the similarity between a candidate and reference text based on the number of matching words between them (Melamed, 1995 [27]).

Another approach, proposed by Rajman and Hartley, 2001 [28], is to give more value to the in-order matching of words. Soon after that, a simplification of that approach was introduced, by Papineni et al., 2002 [29], known as *BLEU*. It is a precision-based measure, and can measure the matching of the candidate text to a set of reference texts by counting the percentage of n-grams overlap between the candidate and reference texts.

Most of the research works on summarization use the standard summarization evaluation metric ROUGE [30] (that stands for Recall-Oriented Understudy for Gisting Evaluation), which determines the summary quality by comparing it to a set of references summaries. This is done by counting the overlapping units (n-grams, word pairs, word sequences) between the generated summary for evaluation and the reference summary (typically created by humans).

Lin, 2014 [30] introduced the following ROUGE scores:

- **ROUGE-N**: based on an N-gram recall between the candidate summary and the set of reference summaries, where N refers to the N-grams length. Therefore some variations of this measure are ROUGE-1 (based on unigrams), or ROUGE-2 (based on bigrams).

- **ROUGE-L**: based on the longest common sub-sequence (LCS). The logic behind this is that the longer the LCS is between two summary sentences, the more similar they are. This is performed estimating the similarity between the candidate and reference summaries.

- **ROUGE-W**: based on the weighted LCS. The length of the consecutive matches is remembered in order to improve the LCS measure, resulting with a weighted LCS (WLCS). This is done to differentiate the LCSes of different spatial relations within the embedding sentences.

- **ROUGE-S**: based on the skip-bigram co-occurrence. As skip-bigram is considered any pair of words in their sentence order, which allows for arbitrary gaps. This measure calculates the overlap of skip-bigrams between the candidate and reference translations. The advantage over the LCS measure is that the skip-bigram will count all the in-order words pairs that match.

- **ROUGE-SU**: this is an extension of the ROUGE-S measure, that takes into account the scenario in which the candidate sentence doesn't have any word pair that co-occurs within the references. Other variants exist of this measure that is based on a different maximum skip distance, for example ROUGE-SU4.

However, the initial implementation of the ROUGE scores doesn't take into account the temporal property of TLS. The improved ROUGE scores for TLS, proposed by Martschat and Markert, 2018 [31], take into account this temporal aspect of the task by aligning the dates in the system and reference timelines. This improved metric will be the baseline for the evaluation criteria for the TLS process in this Thesis work, and will be discussed in depth in the following chapters.

3.4 Pre-trained language models

The textual units that are the base of our work are the *sentences* that are selected to be part of the final summary. However, to select the sentences, first their meaning has to be analyzed, and that is done by analyzing each *word* in the sentence. To be able to process the words by machine learning models, they have to be represented in some form of a numerical representation so that the models can later use it for the calculations.

Word2Vec [32] has proved that a vector (which is constructed by lists of numbers) can be used to represent the words in such a way that their semantic or meaning-related relationships can still be captured (e.g. to be able to recognise if two words are similar, opposites, etc.).
However, each word is always represented by the same vector, no matter in which context it appears in the text.

Many words have multiple meanings depending how they are used in the sentence and its meaning in a given context is important, so the need for *contextualized* word embeddings appeared. Instead of using an unvarying word embedding for each word, these *contextualized* embedding models would take a look at the full sentence before a word embedding is assigned.

The common approach for most of the natural language processing applications in the beginning was to use a recurrent neural network (which uses LSTM - Long Short Term Memory networks), thus requiring a large amount of data, expensive computational resources and many hours of training, while still resulting with poor performance.

An important architecture that moved the focus from RNNs and CNNs (Convolutional Neural Networks) is the *Transformer* architecture (Vaswani, et al., 2017 [33]), that uses an architecture of feed forward networks and attention mechanisms.

On top of this Tranformer architecture, a team of researchers at Google (Devlin et al. 2019 [34]) built *BERT* (Bidirectional Encorder Representations from Transformers), an unsupervised learning architecture. This model outperforms the other models for most of the natural language related tasks.

Most of the deep learning models need a large amount of manually labeled data, which makes them unfeasible in many domains. In such situations a model that can gather linguistic information from some unlabeled data is a great alternative compared to the time-consuming and expensive method of gathering more additional annotation. Due to this, the use of such *pre-trained* word embeddings has showed quite useful for many NLP tasks.

The *BERT* model has been built to pre-train deep bidirectional representations from unlabelled textual data, by conditioning on the left and right context in all the layers. The pre-trained language models have been known for achieving great results for many natural language tasks. They extend the idea of word embeddings by learning the contextual representations from a wide-scaled corpora. *BERT* has been trained with the use of masked language modeling and a task of "next sentence prediction", on a corpus containing over 3330 million words.

The work of Liu and Lapata, 2019 [35] shows how *BERT* can be used also in the text summarization scenario and propose a framework for extractive and abstractive models. In their work they introduced a novel document-level encoder which is based on BERT and is able to express the documents semantic and obtain sentence representations. Their extractive model is constructed on top of the mentioned encoder, by adding to the top several inter-sentence Transformer layers.

Another work that is based on the use of BERT for automatic text summarization is the *Lecture summarization service* proposed by Miller, 2019 [36] which uses BERT model for text embeddings and the KMeans clustering algorithm to identify the sentences closest to the centroid, to be picked for the summary. The purpose is to summarize a lecture content based on a specified summary length.

All of the previously mentioned TLS works are focused on creating timeline summaries in a monolingual environment. Cross-lingual automatic text summarization [37] on the other hand, produces a summary in a language which differs from the source document language. In this scenario, the information from all the available language resources are analyzed to identify the most relevant sentences.

Each of the candidate sentences for the summary is represented as a dense vector, known as sentence embedding. The sentence embeddings are used to compute the similarity and rank the candidate sentences for the summary. One of these language representation models which is adapted to the multilingual scenario is the multilingual variant of BERT called M-BERT (multilingual BERT).

In order to provide a solution for the not-aligned vector spaces between different languages in M-BERT, Sentence-BERT (SBERT) [38] has been introduced. It represents a modification of the pretrained BERT network, that uses siamese and triplet network structures to derive semantically meaningful (close in vector space) sentence embeddings.

Another technique for word representation by a distinct vector is the use of FastText library [39], that is able to learn representations for character n-grams and represent the words as a sum of the n-gram vectors.

Both the SBERT model and the FastText library, using the of aligned word vectors, will be the base of the further experiments in the cross-lingual scenarios.

Chapter 4

Newly proposed dataset

4.1 Dataset

One of the widely used datasets for TLS is the *crisis* dataset[2], that consists of articles, in the English language, focused on long-span events on the armed conflicts: Egypt Revolution, Syria war, Yemen crisis and Libya War (referred to as *Libya, Egypt, Syria* and *Yemen* in this work). The *crisis* dataset, as well as the other standard datasets on which the TLS problem has based on, contains resources in a monolingual scenario, focusing on one single language.

The dataset used for this study has been constructed by collecting multilingual news articles related to these previously mentioned topics, in the following languages of interest: Spanish, French and Italian. These news articles serve as input for the timeline generation during the TLS process. Another important part of the dataset are the reference timeline summaries which serve as ground-truth summaries.

4.2 Ground-truth

The ground-truth summaries, which represent the ideal output, have been constructed manually in order to serve as a good reference point against which we can compare the relevance of the automatically extracted daily summaries. These ground-truth summaries have been created by researching already published timeline summaries for the topics of interest by news agencies, since they have been

carefully picked and produced by professional journalists.

Only the timeline summaries that specify an explicit date (day, month and year) are considered as relevant, so that in the later steps we can have a clear comparison between the reference (ground-truth summary) and timeline (extracted summary) dates.

4.3 News articles collection

4.3.1 News search

For each topic, for every language, a Google search is conducted in order to extract the news articles related to it. This is done using the GoogleNews Python library[1], by passing several arguments:

- *language*: target language in which we want to collect articles
- *time range*: the minimum and maximum date for the article publication. This corresponds to the earliest and latest date that is present in the ground-truth timeline for the topic, expanding them with the range of +/- 10 days (e.g. include articles published also maximum 10 days before the earliest date, or maximum 10 days after the latest date)

4.3.2 Keywords extraction

The search '*phrase*' consists of an array of keywords, that have been extracted from the ground-truth timeline. Two techniques for keyword extraction have been tested:

- *manual extraction*: manually picking the most relevant keywords from the ground-truth timeline that best represent the topic
- *automatic extraction*: using the textacy library[2] that is able to extract key

[1]Available at `https://pypi.org/project/GoogleNews/`.

[2]Available at `https://pypi.org/project/textacy/0.2.3/`.

terms from a document (in our context the ground-truth timeline), with the help of a ranking algorithm

The ranking algorithm used for the automatic keyword extraction task is the *SGRank* algorithm [40] which extracts the top N keywords from a timeline, where N is provided by the user. Several other arguments are provided such as:

- *ngrams*: which ngrams to include as potential keywords (e.g. unigrams and bigrams, phrases containing one or two words will be considered as potential keywords)
- *normalize*: normalize all words in timeline before search for potential keywords (e.g. lemma - return only the base of the word, lower - transform all words to lowercase etc.)
- *include_pos*: pos tags to filter the potential keywords (e.g. noun, adjective)

For the scope of this work, only the unigrams and bigrams will be considered as potential keywords, where first we lowercase all the words in the timeline and extract the top 5 terms belonging to any of the following pos tags: proper noun, adjective or noun. This is done once the raw text of the timeline is tokenized and the proper tag is given to each word in it.

Lower casing the words in the timeline is done so that any ambiguity is avoided between words that have the same meaning but are represented in different case based on the location in the sentence (beginning or middle of sentence).

Table 4.1 shows the comparison between the manual and automatic extraction of keywords for the topic 'Syria' in the French language. It can be seen that using the automatic extraction technique, sometimes the full context of the key term is not preserved. An example for that is the name of the Syrian president, Bashar al-Assad (in French: Bachar al - Assad), often referred to as Assad.
The automatic approach extracted the bigram 'bachar al', although the name in this form doesn't fully specify the president Bashar al-Assad. That's why during the manual extraction we can understand that having as key term 'assad' would provide better results during the news search.

After comparing both keyword extraction techniques, the manual extraction has been chosen, given the fact that it's a supervised way to pick the most relevant aspects of a timeline, focusing on the topic's story and not only on the most common words.

manually extracted keywords	damas,syrie,régime,assad,rebelles
automatically extracted keywords	armes chimiques,barack obama, régime syrien, damas,bachar al

Table 4.1: Comparison between the keyword extraction techniques

4.3.3 Collected news articles

The GoogleNews search returns a list of search results in the format shown in Table 4.2. The important attributes needed for the news articles collection are:

- *date*: the date the article has been published
- *link*: the link of the news article

{ '**title**': '¿Quiénes son los Hermanos Musulmanes?', '**media**': 'El País.com (España)', '**date**': '4 feb 2011', '**desc**': 'En el Egipto de Mubarak no resultaba tan difícil cubrir un mitin de los Hermanos ... La cofradía de los Hermanos Musulmanes (Al Ijuan al Muslimin) es la más ...', '**link**': 'https://elpais.com/internacional/2011/02/04/actualidad/1296774012_850215.html', '**img**': '' }

Table 4.2: GoogleNews search result example

The performed search returns all the available results, but for the sake of simplicity a *number_of_articles* parameter is defined, that is equal to the number of dates present in the ground-truth timeline multiplied by 10.

Once the articles links are available, the news articles are scraped from the web sites using the BeautifulSoup library[3], that is able to scrape information from HTML and XML sites. As most of the important information in an article is contained in the headings (usually containing the articles title or subtitle) and the paragraphs (containing the article body), these elements are scraped for each article. Lastly, each article is saved as a text file in a separate folder that has the article date as a name.

[3]Available at `https://pypi.org/project/beautifulsoup4/`.

4.4 Article preprocessing

The saved articles are first cleaned (from empty or short lines), and later the sentences are splitted and tokenized. After this step, the tokenized articles are saved as a new file, ready for the further processing steps.

4.4.1 Temporal annotation

Each news article has a publication date, but not all of the sentences in the article are referring to that exact date. Some of the sentences can contain a time reference, such as date or time expression (e.g. 'last Sunday'), which does not necessarily coincide with the article's publication date. Therefore, for each sentence, apart from the publication date, a list of reference dates is saved.

The temporal tagging of the sentences is done using Heildeltime[4] [41], a multilingual temporal tagger, able to extract temporal expressions from text and normalize them according to the TIMEX3 annotation standard.

In order to do that, a TreeTagger[5] parameter file is downloaded for each of the resource languages. A TreeTagger is a tool that annotates text with part-of-speech and lemma information, and is needed for the execution of the Heideltime library. After performing the temporal annotation for all the articles, they are saved as new files in the *timeml* format, which is used for temporal annotation of documents.

Each time reference from the article's body is wrapped in a TIMEX3 tag, as shown on the right column of Table 4.3.

Luego, del 29 de enero hasta el 11 de marzo, se llevará a cabo la elección de la Shura, la cámara alta consultiva.	Luego , <TIMEX3 tid="t18" type="DATE" value="2012-01-29">del 29 de enero</TIMEX3> hasta <TIMEX3 tid="t19" type="DATE" value="2012-03-11">el 11 de marzo</TIMEX3> , se llevará a cabo la elección de la Shura , la cámara alta consultiva .

Table 4.3: Temporal annotation example

[4]Available at `https://github.com/HeidelTime/heideltime`.

[5]Available at `https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`.

4.4.2 Corpus objects

Corpus is a collection of linguistic data that consists of large and structured text files, or in our case, annotated articles. Once the data is preprocessed and annotated, it is organized in two data structures, separately for each topic:

- *Tilse corpus*
- *List of dated sentences*

The tilse corpus, created for each of the topics separately, contains a document for each of the article files, where for each sentence the following information is stored in two tuples: (article_publication_date, sentence, time_span) and (referenced_date, sentence, time_span). This object is built using the tilse toolkit[6] (timeline summarization and evaluation).

After that, the list of dated sentences can be obtained from the corpus object and saved separately for each topic. Once the dated sentences are available, the data is ready for the TLS process.

4.5 Dataset characteristics

The timeline format that is used, both for the ground-truth and the extracted summaries, is the same as the timelines in the *T17* dataset. An example of it can be seen in Table 4.4, taken from the ground-truth timeline about the *Egypt crisis*, in Spanish.

As discussed before, the dataset contains resources in 3 languages (Spanish, French and Italian), and for each of them the 4 topics of interest are present: *Libya, Egypt, Syria* and *Yemen*. Table 4.5 gives an overview of the ground-truth summaries that serve as baseline in this study, separately for each language and topic. It can be seen that all of the language resources have different number of dates present in the ground-truth timeline, as well the date ranges can differ during which the given topic has been reported on.

The data structure, that was explained in the previous part, contains the list of dated sentences that serve as candidate sentences for the timeline summaries. We can take a look at the number of available news articles and unique dates from

[6]Available at `https://github.com/smartschat/tilse`.

2013-07-02
El titular de Asuntos Exteriores , Mohamed Kamel Amr , presenta su dimisión . El Gobierno confirma que son cinco los ministros que han abandonado el Ejecutivo en las últimas horas .

2013-07-01
Las Fuerzas Armadas de Egipto dan un ultimátum de 48 horas a las fuerzas políticas para que logren un acuerdo .

2013-06-30
Decenas de miles de personas piden en la plaza Tahrir la renuncia de Mursi .

Table 4.4: Ground-truth timeline excerpt

Lang.	**Topic**	**#gt_dates**	**date_range**	**#avg_sents**
Spanish	Libya	72	16/01/2011-22/08/2011	2
	Egypt	48	25/01/2011-21/08/2013	2
	Syria	27	15/03/2011-04/03/2019	2
	Yemen	34	16/01/2011-27/02/2012	2
French	Libya	28	15/02/2011-20/10/2011	3
	Egypt	59	25/01/2011-08/01/2014	4
	Syria	60	15/03/2011-01/10/2013	2
	Yemen	116	02/02/2011-20/02/2012	3
Italian	Libya	96	15/02/2011-16/02/2015	6
	Egypt	26	11/02/2011-16/08/2013	4
	Syria	53	15/03/2011-23/01/2017	7
	Yemen	210	27/01/2011-27/02/2012	4

Number of ground-truth dates (#gt_dates), the time range of the gt timelines (date_range), and the rounded average sentences per date of each gt timeline (#avg_sents)

Table 4.5: Overview of the ground-truth timelines

the dated sentences present for each topic, separately for each of the language resources, in Table 4.6.

Before proceeding to the TLS process, a *compression ratio* parameter is defined, which will be used to set the number of dates that will be selected for the timeline summary. This parameter is represented as the ratio between the number of dates (#sent_dates) present in the dated sentences, and the dates present in the ground-truth summary (#gt_dates). Table 4.7 gives an overview of the comparison of these two values, for each of the topics and languages.

Lang./Topic		**Libya**	**Egypt**	**Syria**	**Yemen**
Spanish	#art	212	167	311	100
	#date	155	212	331	260
French	#art	121	100	183	100
	#date	202	355	181	270
Italian	#art	85	211	98	52
	#date	330	183	366	138

Number of available news articles (#art) and number of unique dates (#dates) in the dated sentences from the news articles

Table 4.6: Dataset statistics

Topic/Lang.	Spanish		French		Italian		Topic_ratio
	#gt_dates	#sent_dates	#gt_dates	#sent_dates	#gt_dates	#sent_dates	
Lybia	72	155	28	202	96	330	5.63
Egypt	48	212	59	355	26	183	3.50
Syria	27	331	60	181	53	366	6.27
Yemen	34	260	116	270	210	138	1.85
Avg_compr	6.61		4.64		4.50		
Avg_tot	5.25						4.31

Comparison between the number of ground-truth dates #gt_dates and number of unique dates in the dated sentences #sent_dates

Table 4.7: Overview of the date compression ratio

Avg_compr refers to the ratio between *#sent_dates* and the *#gt_dates*, calculated for each language separately. *Topic_ratio* calculates, separately for each topic, the ratio between the values of *#sent_dates* and *#gt_dates*, summed up for all the languages together.

Two values for the *Avg_tot* can be calculated from the previous variables, *Avg_lang* and *Avg_topic*.

Avg_lang is calculated as the average value from the *Avg_compr* values that have been calculated for each of the languages separately, and in our case it equals to 5.25, or rounding it down: 5. *Avg_topic* on the other hand, is calculated as the rounded down average from the *Topic_ratio* values, which equals to 4.

We add an extra experimental value for the date compression ratio, equal to 3, and these parameters will be tested in the later steps to define a ratio that will provide the best results during the TLS process.

Chapter 5

Proposed method

5.1 Problem definition

The TLS problem is defined as follows. Given a set of news articles A, related to a specific news topic T, and a ground-truth timeline gt with an average of k sentences per each date. We declare a number of dates l, defined as $l = s/r$, where s represents the number of source dates and r is the compression ratio. The goal is to build a timeline t that contains l dates and m sentences in total, where $m = l*k$.

This thesis work is based on the date-wise approach, using a graph-based method that takes into account temporal expressions (such as text instances that refer to a specific date) in order to obtain date vectors that will help select the salient sentences from a pool of candidate sentences.

The main steps of the proposed method can be seen on Fig.5.1, and each of them will be explained in greater detail over the next sections.
A point worth mentioning is that Fig.5.1 shows the main steps needed to generate a summary from a collection of news articles, in the simplest scenario when using monolingual resources. If, on the other hand, a multilingual scenario is to be applied, several changes will be done regarding the order of these steps and the input resources for each. This will be discussed better in the section 5.3.

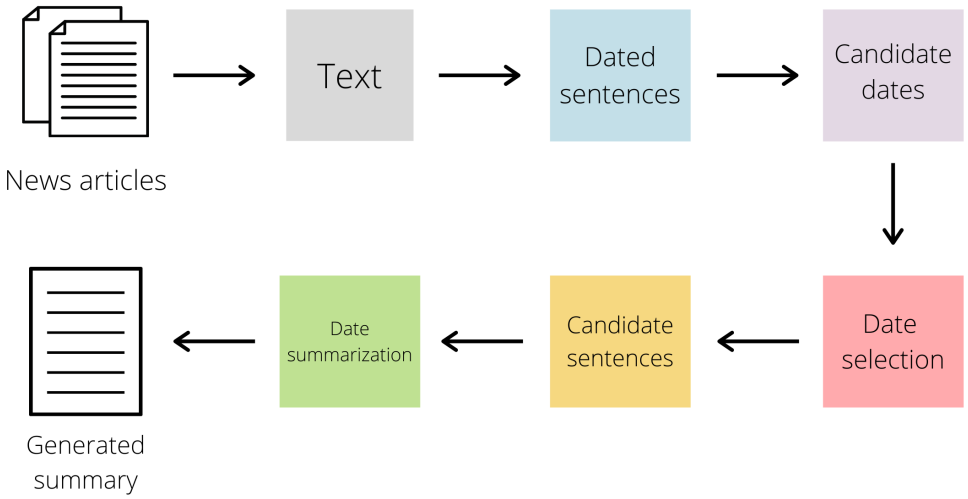

Figure 5.1: Overview of the proposed methodology

5.2 Date-wise approach

Now, let's take a better look at each of the steps shown at Fig.5.1. This process is repeated for each of the topics of interest.

5.2.1 Dated sentences

Having a collection of news articles A as input, a temporal annotation is performed for each of the articles (using Heideltime tool, as explained in the previous Chapter 4), producing as a result a list of temporally annotated (dated) sentences. (steps 1-3 from Fig.5.1)
The timestamp is obtained from either the publication date of the article, or the referenced date in the specific sentence (e.g. 'Last Sunday').

5.2.2 Candidate dates

A set of candidate dates is needed to construct all the available dates for the date selection phase. This set is represented by the collection of the unique dates present in the list of dated sentences. (step 4 from Fig.5.1)

5.2.3 Date selection

Once we have the candidate dates, we need to select only the top l most important ones. (step 5 from Fig.5.1) This is done by performing a graph ranking of the list of input elements, and assigning a score for each of the values in the list. The input list contains the *date* and *text* values, for each item of the dated sentences.

Graph centrality

A *graph* is a representation of relationships between different entities, where these entities are the *nodes* (vertices) of the graph, and the relationships between them are represented by *links* (edges) of the graph.

The concept of *centrality* is very important in order to identify the important nodes in a graph, by telling us how "central" a node is in the graph. Depending on how the importance of a node is defined, we can identify several ranking methods, such as degree, closeness, betweenness centrality, link analysis methods like Pagerank[6] and HITS[8], and many others.

The following graph algorithms have been tested for the date selection phase: *Pagerank*, *HITS* and *Degree*. The implementation of each of them is available in the NetworkX Python package[1].

PageRank computes the ranking of the nodes based on the incoming links structure. Originally, it has been developed as a web site ranking algorithm. The *HITS* algorithm will return two scores for each node: the node's value as estimated by the authorities based on the incoming links, and the node's value based on the outgoing links, estimated by the hubs. *Degree centrality*, in a non-directed graph, is equal to the number of direct connections which a node has with other nodes.

The nodes in the graph can be connected assigning a graph weight, which in our case can be either *i) time*; the difference in time is used as the graph weight between the nodes, or *ii) reference*; using a binary score as weight.

The graph can be either *directed* (there is no symmetry in the edges established between two nodes) or *undirected* (there is a reciprocity in the relationships between two nodes, if A is connected to B, it means that B is connected to A as well). In

[1]Available at `https://networkx.org/documentation/networkx-1.10/index.html`.

our context, the graph is *undirected* and a single weight is assigned to the edge between nodes.

Another argument to the date selection phase is the *threshold*, and all the edges in the graph having a weight lower than the threshold will be removed.

A distance measure can be used to calculate the distance between the dates, and in our experiments we will test the *cosine similarity* measure. This measure calculates the similarity between two vectors by calculating the cosine angle between them.

Selected dates

Once the scores are computed for each date, we need to select the l most salient dates. We have defined previously the value of l as $l = s/r$, where s is equal to the number of unique dates present in the dated sentences, and r is the compression ratio. Before selecting the top l dates, we filter out all the dates that are not within the ground-truth date range.
The result of the Date selection phase is a list of the top l dates from the pool of available dates.

5.2.4 Candidate sentences for summary

In order to create a concise summary for a given date d, we need to choose the candidate sentences to be used as a source. (step 6 from Fig.5.1) Two sets could be considered as the primary source for relevant candidate sentences, as proposed by Ghalandari [1]:

- Pd: Sentences published on or closely after d
- Md: Sentences that mention d

The combination of both sets will be considered in the following experiments, so that all the sentences that have a date which is within the list of *allowed dates* will be considered as candidate sentences.

The list of *allowed dates* is constructed by extending the list of the top l dates with the following dates:

- dates in the range between the earliest ground-truth date and the earliest

article date
- dates in the interval between dates d_{i-1} and d_i, where d_{i-1} and d_i represent two dates in the list of sorted sentence dates

Therefore, the result of this step is a set of candidate sentences for each date d present in the list of *allowed dates*.

5.2.5 Date summarization

The construction of the final timeline is done by combining the summaries for each of the highest l ranked dates. (step 7 from Fig.5.1) For each set of candidate sentences, related to a date d, a summarization method is called. There are several parameters that are considered when constructing the summary.

Each summary should contain a maximum number of sentences #avg_sents, which is a parameter derived from the corresponding ground-truth timeline.

For each of the candidate sentences, a dense vector representation has to be computed, known as *sentence embedding.* In order to do that, a Deep Learning (DL) language model is required, which in our case is a pre-trained Sentence Transformer model[2].
Depending on the task and the linguistic scenario, different pre-trained models are available. For the monolingual approaches in this Thesis work the pre-trained model to be used is a variant of the DistilBERT model (in particular *distilbert-base-nli-stsb-mean-tokens*).

On the other hand, for the multilingual approaches, where the list of candidate sentences contains sentences in several different languages, a multilingual pre-trained model is used which generates aligned vector spaces, which means that similar sentences in different languages will be mapped closer in the vector space. The multilingual models to be used for this scenario are the multilingual distilled version *distiluse-base-multilingual-cased* and the *FastText* models.

As discussed in the previous chapter, Related works, there are different types of summarization methods. In this Thesis work the methods that will be explored are the graph-based, clustering-based and submodular models.

[2]Available at `https://pypi.org/project/sentence-transformers/`.

Graph-based models

Applying a graph-based model to the natural language scenario includes the following steps:

- Identify the text units (in our case sentences) and add them as nodes in the graph.
- Identify the relations connecting these text units, and use them to draw links between the nodes in the graph. The links can be directed or undirected, weighted or unweighted.
- Iterate the graph-based ranking algorithm of choice until it converges.
- Rank the nodes based on their score.

As mentioned before, *TextRank*[7] is one of the most popular graph-based models used for extractive summarization.
To apply *TextRank*, a graph is built where a node is added to it for each of the sentences present in the candidate sentences. To define the relation between the sentences in a graph-based model, different measures can be used: words overlapping, cosine distance or query-based similarity.

The *TextRank* method is based on ranking the sentences based on their scores which are returned by running the *PageRank* algorithm on a graph of pairwise sentences similarities. This method determines the similarity relation based on the content they share, calculated as the number of common tokens between two sentences divided by the length of each, so that long sentence promotion would be avoided.

The similarity function for two sentences S_i, S_j, represented by a set of n words (in S_i as $S_i = w_1^i, w_2^i ..., w_n^1$), is defined as:

$$\text{Sim}\,(S_i, S_j) = \frac{|\{w_k \mid w_k \in S_i \cap w_k \in S_j\}|}{\log\,(|S_i|) + \log\,(|S_j|)} \tag{5.1}$$

The result of applying the similarity function is a dense graph that represents the document collection. After this graph is computed, the *PageRank* algorithm is used to compute each node's importance, and finally the most important sentences are selected.

The following graph-based summarization models have been tested during the future steps:

- TextRank algorithm (gensim[3] re-implementation, Sumpy[4] and Sumy[5] implementations)
- LexRank [36] algorithm that computes the importance of a sentence based on the concept of eigenvector centrality within a graph representation of sentences (Sumpy and Sumy implementations)
- CoreRank [42] (implementation[6])
- PacSum [43] (implementation[7])

Given the fact that TextRank method is language indipendent, for the multilingual approaches the graph-based summarization model that will be tested in an implementation of the PageRank algorithm using SentenceTransformers.

The nodes in the graph are represented with the candidate sentences for the summary, while the edges which are connecting them are assigned a weight which is calculated based on the sentence embeddings. After creating the graph, the PageRank algorithm is ran and the top sentences to be included in the summary are selected.

Clustering-based models

As mentioned in the previous chapter, the *centroid-based* methods are quite common approach for the extractive text summarization. These methods rank the sentences based by their similarity to the centroid of all the sentences. A common measure to be used to calculate the similarity is the *cosine similarity*.

Cosine similarity is used to measure how close two vectors (in our context two sentences) A and B are, based on their angle, and it can be defined as follows:

$$\text{sim}(A, B) = \frac{A \cdot B}{|A||B|} \tag{5.2}$$

The summary is constructed by selecting the top ranked sentences from the ranked list of sentences in a decreasing order, until the length of the summary is

[3] `https://radimrehurek.com/gensim_3.8.3/summarization/summariser.html`

[4] `https://github.com/kedz/sumpy`

[5] `https://github.com/miso-belica/sumy`

[6] `https://github.com/MorenoLaQuatra/summarization_collection`

[7] `https://github.com/mswellhao/PacSum`

equal to the defined summary length parameter.

Several implementations of the centroid-based summarizatio methods will be evaluated during the score of this Thesis work, among which are:

- Sumpy[8] implementation
- Centroid-Rank [10] implementation, ranking the sentences based on their similarity to the centroid of all sentences
- Centroid-Opt [11] implementation, greedily optimizing a summary to be similar to the centroid of all the sentences

Submodular models

Submodular [13] models represent another option for the extractive text summarization. These type of models guarantee that there is a greedy algorithm for monotone submodular function maximization, where it's guaranteed that the obtained summary solution, $\widetilde{S}$, is almost as good at the best possible solution, S_{opt}, based on an objective $\mathcal{F}$.

The appliance of sumbodularity in text summarization can be defined as follows: Let's define a *ground set* V that consists of all the sentences in the document collection. The task of extractive text summarization is to choose a subset $S \subseteq V$ to represent the ground set V.
Since S represents a summary, usually its length is limited by a predefined parameter.

The implementation of this type of summarization method that has been used in this Thesis work is the one by Lin and Bilmess, 2011 [13]. This work proposed several constraints on S that can be used, among which are the *knapsack constraints*, defined as: $\sum_{i \in S} c_i \leq b$ for c_i as the non-negative cost for selection of unit (sentence) i and b as our *budget*.
If a set function ($\mathcal{F} : 2^V \Longrightarrow \mathbb{R}$) is used to measure the summary set S quality, the summarization problem can be defined as a combinatorial optimization problem as follows: *Find*

$$S^* \in \underset{S \subseteq V}{\operatorname{argmax}} \mathcal{F}(S) \text{ subject to: } \sum_{i \in S} c_i \leq b \tag{5.3}$$

[8] `https://github.com/kedz/sumpy`

5.2.6 Generated summary

The final result of the previously described steps, is a predicted timeline, containing the daily summaries for the most important dates from the collection. (step 8 from Fig.5.1)

The format of the predicted timeline is the same as the one mentioned in the previous section, *TL17 format*, in order to have a fair comparison to the ground-truth timeline.

An example of a predicted summary for the *Egypt* topic in the Spanish language, using the TextRank algorithm, can be seen in Table 5.1.

2012-12-25 Gigi Ibrahim : " La inestabilidad en Egipto está haciendo que haya gente que pida la vuelta de Mubarak " La joven activista , considerada por ' Time ' como una de las líderes de Tahrir , cree que la nueva Constitución egipcia no tiene legitimidad Egipto aprobó el pasado 25 de diciembre su primera Constitución tras la caída del rais Hosni Mubarak .
2013-01-01 Estuvo de más sin embargo el bienintencionado optimismo , que a partir de lo sucedido en Túnez y en Egipto apostó por la teoría kissingeriana del dominó , profetizando que un país tras otro derrocaría a sus autócratas para implantar la democracia . La función del Consejo de los Expertos iraní es transferida como órgano consultivo " en materias pertenecientes a la ley islámica " al centro islamista por excelencia , la Universidad de al - Azhar .
2013-01-25 La ' primavera árabe ' impone un nuevo equilibrio de poderes en Oriente Próximo La oposición hace una demostración de fuerza en Egipto dos años después de la revolución Los grupos islamistas y seculares pugnan por el poder en la zona Después de dos años de revolución , nada parece seguro en Oriente Próximo . Los partidos y movimientos no islamistas han convocado mañana viernes manifestaciones en todo el país en protesta por el rodillo de la Hermandad , la agrupación a la que pertenece el presidente Mohamed Mursi , y sus aliados salafistas (rigoristas) .

Table 5.1: Predicted summary excerpt

5.3 Applied methodologies

The TLS process is implemented by testing it for several proposed methodologies:

- **Single language**, perform TLS separately for each of the language resources
- **Multilingual date model**, consider all the source dates for the date selection phase
- **Early translation**, translate all the resources to a given target language T before performing date selection and date summarization
- **Mid translation**, summarize the single language resources, translate the summaries to target language T and perform date selection and "further" summarization
- **Late translation**, perform date selection for all the single language resources and perform cross-lingual summarization, construct final summary

Each of these approaches is explained more in depth as follows:

5.3.1 Single Language

In this scenario, the TLS process is done for each of the language resources separately.

The date selection task takes as input the list of dated sentences for the specific language, and assigns a score to each of the dates. After the dates are ranked, the top l dates (for $l = s/r$, and for s we take the number of unique dates for the given language) are selected for the following step.

The summarization is perfomed using the chosen summarization method, in our case after performing a grid search between different methods, as mentioned, the *TextRank* algorithm has been chosen.
The Deep Learning model used for the sentence embeddings is the DistilBERT model (distilbert-base-nli-stsb-mean-tokens)[9], given the fact that this task is monolingual.

After performing the summarization step, a predicted system timeline is built, and compared with the ground-truth timeline for the corresponding language. An example of the date selection and summarization scores, computed for the French language, separately for each topic, is shown in Table 5.2.

[9]Available from `https://pypi.org/project/sentence-transformers/`.

	concat		agree		align+m:1		Date sel.
Topic	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Libya	0.3856	0.1256	0.0537	0.0053	0.0904	0.0084	0.3037
Egypt	0.4958	0.1528	0.0386	0.0072	0.0631	0.0102	0.1891
Syria	0.5124	0.1926	0.0369	0.0121	0.0723	0.0215	0.1886
Yemen	0.5652	0.1950	0.0925	0.0209	0.1341	0.0255	0.2934
AVG	**0.4897**	**0.1665**	**0.0554**	**0.0114**	**0.0900**	**0.0164**	**0.2437**

Table 5.2: Example of date summarization and date-selection scores

5.3.2 Multilingual Date Model

The *Multilingual date model* approach, uses for the set of dates all the available dates for all the languages present. This gives an opportunity that the set of dates for a given language is extended also to the dates that might not be present for that language, but are in the other languages.

Each of the languages has a set of dates, that contains a certain number of dates and has a specific date range (earliest and latest date in the ground-truth timeline). An example of that is shown in Table 5.3, where the number of dates and the date range is given for each of the languages for the topic *Libya*, as well as the *multilingual* number of dates and date range when all the languages are combined.

Language	#dates	date_range
Spanish	155	16/01/2011 - 22/08/2011
French	202	15/02/2011 - 20/10/2011
Italian	330	15/02/2011 - 16/02/2015
Multilingual date model	496	16/01/2011 - 16/02/2015

Table 5.3: Multilingual Date Model example

From the Table 5.3 we can see that the date range used for the Multilingual date model is the one that includes the earliest and the latest date present when all the language resources are joined into one set. The number of dates available for the date selection in this approach, is the number of unique dates present for all the languages together.

Here again we apply the same reasoning, we need to extract the top l dates from this set of s dates. This is done with the use of the compression ratio r, so that the number of dates to be chosen from the date selection phase for the topic

Libya using the *Multilingual date model* at the end equals to 124. Table 5.4 shows the number of dates to be selected during the date selection for the different topics, for the approaches that have been discussed so far.

Approach	Libya	Egypt	Syria	Yemen
Single language (Spanish)	39	53	83	65
Single language (French)	51	89	46	68
Single language (Italian)	83	46	92	35
Multilingual date model	124	130	175	109

Table 5.4: Date selection statistics

The date selection phase is the same for all of the languages in this approach, since the set of dates contains all the source dates available for the three languages of interest.

Since the summarization will be done in a monolingual scenario, using only the resources of each language separately, the *TextRank* algorithm is used with the Deep Learning model DistilBERT, same as for the *Single language* approach.

In terms of date selection scores, this *Multilingual date model* approach on average outperforms the *Single language* approach, with the biggest improvement for the *Italian* language. The more detailed results will be discussed in the following Chapter 6.

5.3.3 Early Translation

In the *Early Translation* approach, the goal is to merge the languages together in order to have richer resources for a given target language. The concept of Early translation is the following: translate the resources from a source language S to a target language T, and use this extended resource set as the baseline for the timeline summarization for the language T. An overview of this process is shown on Fig.5.2.

For the date selection phase the *Multilingual date model* is used, in order to expand the date range and also the number of dates to be selected for the target language.

There are two possibilities for the process of enriching a target language's resources with other languages:

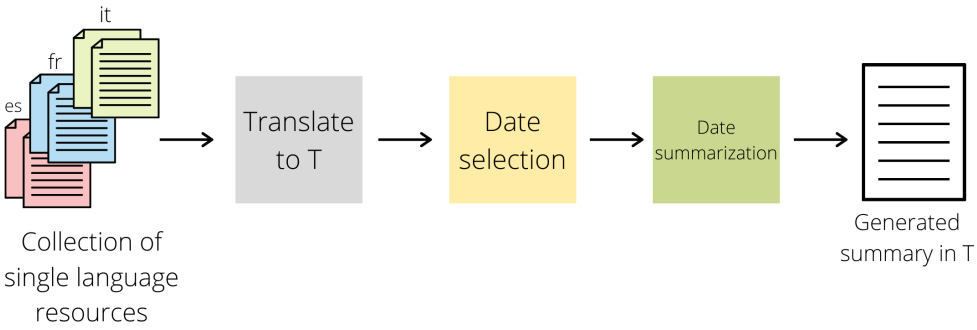

Figure 5.2: Overview of the Early Translation approach

- merge two languages together (source language and target language)
- merge three languages together (two source languages and one target language)

In the first case, a target language T is enriched by translating the resources of one source language S to T, and performing summarization of these expanded resources, for the target language T. The latter one instead translates the resources of the remaining two source languages S_1 and S_2 to a target language T and performs the summarization for T.

The translation of the resources is done using a free Python API for Google Translate[10]. The text units to be translated are the dated sentences for the source language(s), passing as an argument the target language T.

The result of the translation of a Spanish sentence to Italian as a target language, can be seen in Table 5.5.

2011-06-02 El 2 de junio escuchó impasible su condena a cadena perpetua por estos crímenes .	2011-06-02 Il 2 giugno ha ascoltato impassibile la sua condanna a vita per questi crimini.

Table 5.5: Candidate summary excerpt: before and after translation

Since all the resources are in one single language, the target language T, again the Deep Learning model used for the sentence embeddings is the DistilBERT model.

[10]Available at `https://pypi.org/project/google-trans-new/`.

5.3.4 Mid Translation

This approach changes the order the two phases of the TLS process, by first applying the summarization on the resources for all the languages separately by maintaining all the dates, translates these summaries to a given target language T and at the end perform the date selection phase and (if needed) a further summarization. This process is pictured on Fig.5.3, where the * refers to the possible further summarization step that may be added.

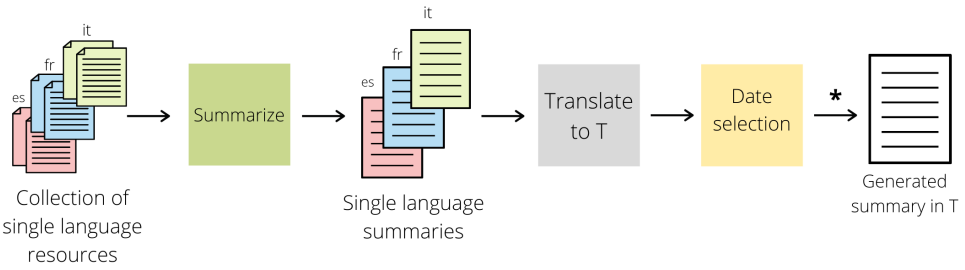

Figure 5.3: Overview of the Mid Translation approach

At the beginning each of the language resources are summarized separately, using TextRank summarization method with the DistilBERT sentence embedding model, without performing any date selection beforehand, so all of the available dates are present in these summaries. After this process, we have a separate summary timeline for each of the languages. These timelines may have a *candidate* summary in several languages for the same date.

Next, these summary timelines are all translated to a target language T, therefore constructing one joint summary containing for each date all the possible daily summaries that were available, translated to T.

Finally, the date selection phase can be performed to this joint summary, in order to remove the date summaries for the dates which will not be among the top l dates. As previously, the *Multilingual date model* has been used for this phase, expanding the pool of dates available for the selection.

Since the length of these final summaries after the date selection could be longer than the parameter that is defined as the average summary length (#avg_sents parameter derived from the ground-truth timeline), and could therefore contain unnecessary information, a further summarization technique has been also proposed.

In the following Chapter 6 the detailed results of the *Mid Translation* approach

are discussed, as well as the advantages of using a further summarization step during this approach.

5.3.5 Late Translation

The last approach that was implemented and tested is the *Late Translation* technique, that deals with resources of different languages at the same time, and can be seen on Fig.5.4.

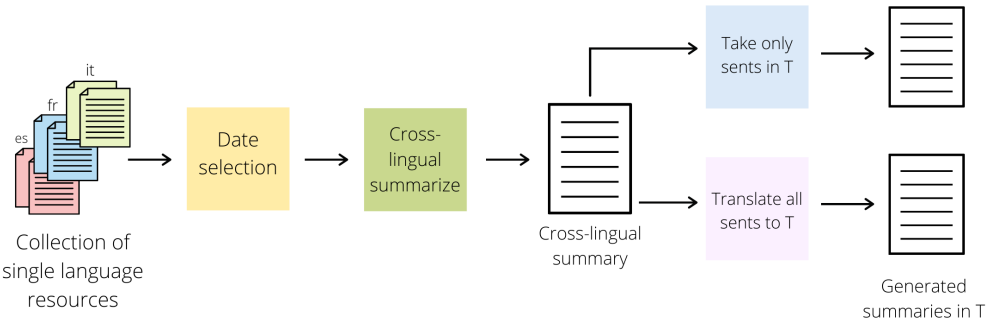

Figure 5.4: Overview of the Late Translation approach

This process starts with the date selection phase, using the *Multilingual date model*, and using these selected dates a universal resource set is created with the sentences dating to those dates, in all the three different languages.

After this, a cross-lingual summarization is performed. Since the resources are not in a single language, but instead in two or three different languages, a multilingual Deep Learning model is needed to construct the aligned sentence embeddings. As mentioned before, two models will be tested for this purpose.

The first is the multilingual version of the Sentence Transformer model (distiluse-base-multilingual-cased[11]) which will be used for that purpose due to the fact that the vector spaces between the languages are aligned and the same words will be closer in vector space even if they are in different language.

The second option that will be tested is the use of the aligned word vectors by FastText[12], where an aligned word vector is available for each of the languages

[11] Available at `https://www.sbert.net/examples/training/multilingual/README.html`.

[12] Available at `https://fasttext.cc/docs/en/aligned-vectors.html`.

of interest in this Thesis work, and can be used for the sentence embeddings representation.

Both of these models can deal with sets of sentences that are in different languages, and represent them accordingly. The summarization methods that will be tested for this approach are the implementations of the Submodular (Lin and Bilmes, 2011 [13]) and Centroid (Radev et al., 2004 [10], Ghalandari 2017 [11]) algorithms.

Given the fact that the input sentences to the summarization are in various languages, also the output summary contains sentences in one or many languages. An example of this can be seen in Table 5.6, where for one date there are sentences in more than one language.

2011-08-05
Così il 5 agosto Muahamar Gheddafi , asserragliato nel suo bunker , avrebbe scritto una missiva a Silvio Berlusconi per chiedere di “ fermare i bombardamenti che uccidono i fratelli libici e i bambini ” .
L’ ex - dirigeant libyen Mouammar Kadhafi a écrit le 5 août à son " ami " le chef du gouvernement italien Silvio Berlusconi pour lui demander d’"arrêter les bombardements qui tuent nos frères libyens et nos enfants " , selon une lettre publiée lundi 24 octobre sur le site du magazine " Paris Match " .
2011-08-22
El dictador libio siembra el desconcierto entre los jefes de los sublevados con una guerra de propaganda .
Los rebeldes , con pocos medios , se ven obligados a replegarse a Bengasi .

Table 5.6: Cross-lingual summary excerpt

For the computation of the final summary there are two scenarios that will be tested and evaluated:

- Use all the available sentences, translated in target language T
- Select only the sentences that are in the target language T

The comparison between these two scenarios, as well as the performance of the *Late Translation* approach in general when compared to the previously described approaches, is discussed in the next chapter.

5.4 Implementation tools

The programming language in which this Thesis work has been developed is *Python*. The following libraries have been used for the implementation of the different concepts in this project:

- *spaCy* [44]: a free open-source library that is used for Natural Language Processing tasks in Python. This library has been used for the processing of the text in the ground-truth timelines using the appropriate language model.
- *textacy* [45]: library built on Spacy, used in this project for the keyterm extraction.
- *GoogleNews* [46]: a free library used to retrieve Google News results based on a given query. Used in the data collection process for gathering urls of the news articles for the topics based on the keyterms.
- *BeautifulSoup* [47]: library that allows to scrape information from Web pages. Used in this project for the collection of news articles by scraping the gathered urls.
- *tilse* [48]: toolkit used for timeline summarization and evaluation. Used for the creation of corpus objects, representation of the timelines in the appropriate format, evaluating of timelines using ROUGE metrics etc.
- *Google Translate* [49]: a free Python API for Google Translate. Used for the translation of the resources between languages.
- *Sentence Transformer* [50]: a framework providing an easy computation of the setence embeddings. This framework has been used for sentence embeddings representation in the multilingual approaches.
- *FastText* [51]: a library for text classification and representation learning. Used in the multilingual approaches for the sentence embeddings computation using the aligned word vectors for the languages of interest.
- *NetworkX* [52]: is a Python package used for creating, manipulating, and studying the structure and functions of complex networks. Used in this Thesis work for the graph representation and ranking algorithms application.
- *Scikit-learn* [53]: a machine learning library for Python. In the scope of this project used for the similarity computation, preprocessing functions for the vectors etc.

- *Numpy* [54]: one of the fundamental packages for scientific computing. It is used for the convertion of input data to vector, as well as performing some basic math operations.
- *Gensim* [55]: a library used for modelling of topics, retrieve similarity and document indexing with large corpora. Used in the implementation of the TextRank algorithm.

Chapter 6

Experiments

6.1 Experimental design

6.1.1 Summarization evaluation

The quality of the extracted summaries is evaluated using the standard summarization evaluation metric, *ROUGE*. This metric is commonly used to evaluate the system (predicted) summary s with respect to the set of reference (ground-truth) summaries R. The most popular variant is the ROUGE-N which measures the overlap of the ngrams between the system and reference summaries. A more detailed explanation of the ROUGE-N metric is given as:

"For a summary c, let us define the set of c's N-grams as $ng(c)$. $cnt_c(g)$ is the number of occurrences of an N-gram g in c. For two summaries c_1 and c_2,

$$cnt_{c1,c2}(g) = min\{cnt_{c1}(g), cnt_{c2}(g)\}$$

is the minimum number of occurrences of g in both c_1 and c_2." [31]

ROUGE-N F_1 score is equal to the harmonic mean of recall and precision, which are defined as shown below:

$$rec(R, s) = \frac{\sum_{r \in R} \sum_{g \in ng(r)} cnt_{r,s}(g)}{\sum_{r \in R} \sum_{g \in ng(r)} cnt_r(g)} \tag{6.1}$$

$$prec(R, s) = \frac{\sum_{r \in R} \sum_{g \in ng(s)} cnt_{r,s}(g)}{|R| \sum_{g \in ng(s)} cnt_s(g)} \tag{6.2}$$

The *recall* in our context can be defined as the fraction of ngrams that are present both in the reference summary r and system summary s, out of all the ngrams in the reference summary r, while *precision* is defined as the fraction of the ngrams present in both r and s, out of all the ngrams present in the system summary s.

The evaluation of the quality of the system timelines is done using the most common ROUGE-N metrics used for TLS evaluation, which are ROUGE-1 and ROUGE-2. The reference summaries set R in the scope of this work contains of only one timeline summary.

The following ROUGE implementation methods have been tested in this thesis work:

Concatenation-based ROUGE

This method[1] (referred to as *concat* in the following text) concatenates items from timelines and runs ROUGE.

Having a timeline $t = (d_1, s_1), ..., (d_k, s_k)$, the summaries s_i are concatenated and a document *s'* is produced, discarding the temporal information. The same transformation is done for both the system and reference timelines, and the ROUGE is used on the newly constructed documents.

Date-agreement ROUGE

This technique[2] will be referred to as *agree* in the following text.

Having a reference timeline r and system timeline s, *r(d)* contains the summary of date d and *s(d)* contains the summary of d. *s(d)* can be possibly empty if d is

[1]Discussed in the following works: [56][57][58][23]

[2]Discussed in the following works: [4][22]

not included in the timeline. The recall for a date d is defined as follows:

$$rec(d, R, s) = \frac{\sum_{r \in R(d)} \sum_{g \in ng(r)} cnt_{r,s(d)}(g)}{\sum_{r \in R(d)} \sum_{g \in ng(r)} cnt_r(g)} \tag{6.3}$$

By extending the recall *rec(d,R,s)* to the set of dates D_R, the formula for the ROUGE recall looks like this:

$$rec(R, s) = \frac{\sum_{d \in D_R} \sum_{r \in R(d)} \sum_{g \in ng(r)} cnt_{r,s(d)}(g)}{\sum_{d \in D_R} \sum_{r \in R(d)} \sum_{g \in ng(r)} cnt_r(g)} \tag{6.4}$$

The same reasoning is used for the ROUGE precision formula, where we average with respect to D_s instead of D_R. This approach also considers the temporal information by evaluating the predicted summary for each day separately, but it requires that the dates in the system and reference match exactly.

Date-alignment ROUGE

The following technique[3] will be referred to as *align+m:1* in the following text, and it improves the previous metrics by considering the temporal and semantic similarity of the daily summaries, and does not require an exact match between the dates. This is done due to the fact that summaries that are close in time could still be relevant for the given topic.

This method first aligns the dates both in system and reference timelines, and after computes the ROUGE scores.

This metric has been defined as:

"Let R be a set of reference timelines and let s be a system timeline. The proposed alignment-based ROUGE recall relies on a mapping:

$$f : D_R \rightarrow D_s$$

[3]Proposed in the following work: [31]

that assigns each date $d_r \in D_R$ in some reference timeline, a date $d_s \in D_s$ in the system timeline.

The penalization of the date difference while comparing the summaries is done assigning a weighting factor $t_{dr,ds}$ (6.5) to each date pair d_r, d_s. Only the *weighting factor* that is based on the difference in number of days between d_r and d_s will be considered."

$$t_{dr,ds} = \frac{1}{|d_r - d_s| + 1} \tag{6.5}$$

Finally, the ROUGE alignment-based recall rec(R, s, f) can be defined as:

$$\frac{\sum_{d \in D_R} t_{d,f(d)} \sum_{r \in R(d)} \sum_{g \in ng(r)} cnt_{r,s(f(d))}(g)}{\sum_{d \in D_R} \sum_{r \in R(d)} \sum_{d \in ng(r)} cnt_r(g)} \tag{6.6}$$

For the precision formula the following alignment is considered instead:

$$f : D_s \to D_R$$

In order to compute the date alignments, every date pair $(d_r, d_s) \in D_R X D_s$ is associated with another value, defined as *cost* $c_{dr,ds}$ of assigning d_r to d_s.

Depending on the cost, the following alignments are possible:

- *Date alignment:* the cost only depends on the date distance while not considering the semantic similarity. The cost is defined as follows:

  $$c_{d_r,d_s} = 1 - \frac{1}{|d_r - d_s| + 1} \tag{6.7}$$

  It is required that the alignment is injective, or if $|D_R| > |D_s|$, some $d_r \in D_R$ will be not aligned.

- *Date-content alignment*: includes in the cost also the semantic similarity, and an approximation of it is represented by the ROUGE-1 F_1 score between two summaries:

  $$c_{d_r,d_s} = (1 - \frac{1}{|d_r - d_s| + 1) * (1 - R1(d_r, d_s))} \tag{6.8}$$

  For $R1(d_r, d_s)$ as the ROUGE-1 F_1 score that compares the reference summary for date d_r with the system summary for d_s. This alignment is also injective.

- *Many-to-one Date-content alignment*: the injectivity alignment is dropped for this metric. The date assignment is done using a greedy algorithm, by choosing for every date in D_R a date in D_s for which the cost would be minimal.

Since for the initial implementation the cost only depends on the date distance while not considering the semantic similarity, the improved version called *Many-to-one Date-content Alignment* will be used instead.

For the evaluation of the generated summaries the ROUGE *R-1* and *R-2* scores will be used, in the three different forms (concatenation, date agreement and date alignment), using the *tilse*[4] toolkit for timeline summarization and evaluation.

6.1.2 Date selection evaluation

A date *F1-score* is used to compare the dates between the system and reference timelines. The formula of it is given as shown with the given formula:

$$F_1 score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.9}$$

Precision and recall are defined as follows:

$$Precision = \frac{dates_{s,r}}{dates_r} \tag{6.10}$$

$$Recall = \frac{dates_{s,r}}{dates_s} \tag{6.11}$$

For:

- $dates_{s,r}$: number of dates which are present both in system and reference timeline
- $dates_r$: number of dates in reference timeline
- $dates_r$: number of dates in system timeline

To evaluate the performance of the different approaches that have been proposed for the TLS process, the results of each of them will be discussed in the section 6.3, as well as the comparison between them.

[4]Available at `https://github.com/smartschat/tilse`.

6.2 Configuration settings

Several settings have been established after testing different possible parameters for the following methods:

6.2.1 Compression ratio

As mentioned previously, the values to be tested for this parameter are: 3, 4 and 5. An example of the comparison between the results, averaged for all the topics, for the different values of this parameter, can be seen in Table 6.1.

Setting the compression ratio to 5 produces the worst results concerning the date-selection aspect, due to the significantly lower number of dates to be selected. The summarization scores are slightly better due to the lower amount of irrelevant information in the summaries. A compression ratio equal to 3 on the other hand, results with the highest date-selection scores among the 3 scenarios, although the summarization scores are significantly lower.

As a result, in order to have the best results on average, concerning both the summarization and date-selection aspects, a compression ratio of 4 has been selected.

	concat		**agree**		**align+m:1**		**Date sel.**
	R1	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
r=5	0.4976	0.1630	0.0532	0.0100	0.0881	0.0147	0.2386
r=4	0.4898	0.1664	0.0554	0.01142	0.0901	0.0165	0.2435
r=3	0.4622	0.1646	0.0592	0.0113	0.0934	0.0160	0.2704

Table 6.1: Comparison between compression ratio values

For the different approaches that have been described in Chapter 5, a varying number of dates are present for the Date selection phase. This is due to the fact that when several language resources are combined, the set of available dates for this phase is extended.

Therefore, as mentioned before, the number of dates l available for the date selection phase, is equal to the ratio between the number of source dates s and the compression ratio r. For the same target language and topic this number can vary depending on the specific approach to be employed.

6.2.2 Date-wise approach parameters

The following parameters have been selected as the best performing for the date-selection process:

- Using *reference* as a graph weight connecting two nodes in the graph. For the *reference* technique, a weight is added for each pair of source (the published date of the article) and referenced date (the date the specific sentence is referring to). This has shown as better technique when compared to the *time* technique where the weight is assigned as a difference in time.

- Setting *threshold = 0.9*. The nodes in the graph are connected and are being assigned a given weight which represents their similarity. The threshold value can be used to limit the weight between two nodes (two dates) in order to be considered part of the graph, by removing them if their weight is lower than this value. After experimenting with different values for the threshold, 0.9 has shown to obtain the best results for the date selection.

- Use of *PageRank* algorithm for the computation of the nodes ranking. This algorithm has outperformed the *HITS* and *Degree* algorithms, having a positive effect on the date selection score.

6.2.3 Summarization method

Monolingual tasks

In the previous section, the list of summarization methods that have been tested in the scope of this work has been defined. Among all of them, *TextRank* has shown to be the best performing algorithm in the monolingual scenarios, and gave, on average, the best results among the other algorithms, or however results relatively close to the best performing ones. This algorithm is language independent, which is why it's not appropriate for the multilingual scenarios, that will be discussed later on.

BM25/Okapi-BM25 [59] is one of the ranking function that can be used in the *TextRank* implementation. *BM25* represents a variation of the TF-IDF model (Term Frequency–Inverse Document Frequency), which uses a probabilistic model.

For two sentences R, S it can be defined as follows:

$$BM25(R,S) = \sum_{i=1}^{n} IDF(s_i) \cdot \frac{f(s_i,R) \cdot (k_1+1)}{f(s_i,R) + k_1 \cdot \left(1 - b + b \cdot \frac{|R|}{avgDL}\right)} \tag{6.12}$$

where k and b and defined parameters and $avgDL$ is the average sentence length in the collection.

Since the function implies that if a word appears in more than half of the documents from the collection, it will have a negative value, the following formula has been used for correction:

$$IDF(s_i) = \begin{cases} \log(N - n(s_i) + 0.5) - \log(n(s_i) + 0.5) & \text{if } n(s_i) > N/2 \\ \varepsilon \cdot \text{ avg } IDF & \text{if } n(s_i) \leq N/2 \end{cases} \tag{6.13}$$

where ε is a value betwwen 0.5 and 0.3 and $avgIDF$ is the average IDF for all the terms.

The *BM25* ranking function has been used for the *TextRank* implementation in the date summarization phase.

Multilingual tasks

Since TextRank is not language dependent, other summarization methods need to be used for the tasks that deal with resources in several different languages.

For this scenario, a Deep Learning language model is needed to represent the sentences from different languages in an aligned manner, so that two words in different languages that have same meaning will be closer in the vector space to each other.
As mentioned before, the models to be used are the SBERT and FastText models, and the results of both will be compared in the following section.

Regarding the summarization algorithm, among the methods that have been tested, the implementation of the Submodular summarization method has proved to be the best performing one on average, and it will be used for the demonstration of the results.

6.2.4 Experimental settings

For each of the languages and topics of interest, the following settings are followed for the scope of the experiments:

- A set of dated sentences is available for the each of the topics of interest, as well as a ground-truth timeline to serve as reference for the summarization evaluation. The demonstrated results in the next section are averaged over all topics for each of the languages.
- The l most important dates are selected from the set of available dates from the dated sentences, and for each of them the candidate sentences are collected. The number of dates to be selected is calculated as described in section 6.2.1.
- For each of the most important dates a summary is built with a maximum length of k sentences, resulting with a final summary of length $m = l * k$.

6.3 Results

As introduced in Chapter 1, the main objective of this work is to employ the multilingual resources in order to combine the knowledge extracted from them and improve the quality of the generated summaries. In order to do that, different approaches have been proposed, and the results of each of them will be demonstrated in this section.

Two aspects can be improved for the generated summaries:

- *date summarization scores*, evaluated using the ROUGE metric
- *date selection score*, measured by the F_1-score

The proposed approaches that have been described in Chapter 5 are the following: *Single Language, Multilingual Date Model, Early Translation, Mid Translation*, and *Late Translation*.
The Single Language approach evaluates the generated summaries for each of the languages separately, while the Multilingual Date Model aims to improve the date selection aspect mentioned previously. The attempt to improve the date summarization scores is carried out using the various translation techniques that have been proposed.

In the next sections, for each of these approaches the following arguments will be covered:

- Presentation of the results obtained employing the specific approach
- Discussion of the effects of the approach compared to the previous approaches

6.3.1 Single Language

The averaged results on all the topics, using the *Single language* approach, for the different languages, are shown in Table 6.2.

	concat		**agree**		**align+m:1**		**Date sel.**
Language	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Spanish	0.3596	0.1495	0.0752	0.0441	0.0968	0.0474	0.2419
French	0.4897	0.1665	0.0554	0.0114	0.0900	0.0164	0.2437
Italian	0.3470	0.0879	0.0143	0.0013	0.0301	0.0029	0.1324

Table 6.2: Single Language: summarization and date selection scores

Regarding the date selection scores, we can see that the Spanish and French language resources perform quite similarly, while the Italian language results with a much lower date selection F_1 score when compared to the other languages.

As mentioned before, the Italian language has the lowest number of resources, which can be an explanation for the significantly lower score during the date selection phase.

Taking a look at the summarization ROUGE scores, we can see that the French language has significantly higher scores compared to the other two languages, especially when considering the *concat* metric. Taking into account the other metrics, *agree* and *align*, the scores of the Spanish and French resources are quite closer, while the Italian ones are much lower.

6.3.2 Multilingual Date Model

Next, we will take a look the the results implementing the *Multilingual Date Model*, which can be seen in Table 6.3. The results are averaged over all topics, for the different languages of interest.

After applying this technique before the date selection phase, we can see that the F_1 score for the date selection is quite similar for the three languages, in particular benefiting the Italian language. A detailed overview of the effect of this date model on the date selection score can be seen on 6.1.

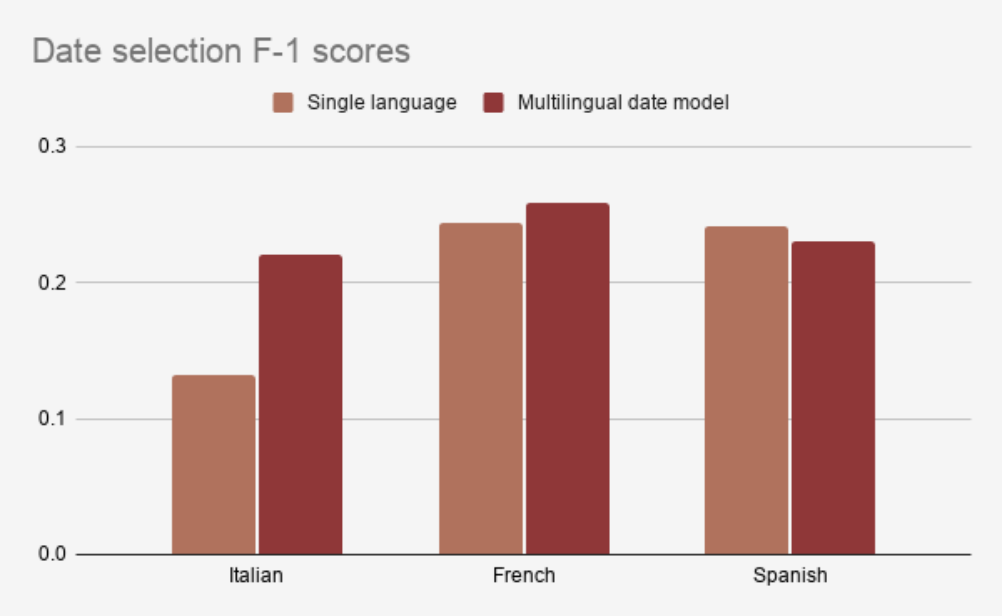

Figure 6.1: Effect of Multilingual Date Model on date selection score

	concat		agree		align+m:1		Date sel.
Language	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Spanish	0.2731	0.1289	0.0707	0.0339	0.0904	0.0369	0.2305
French	0.4145	0.1467	0.0538	0.0099	0.0863	0.0143	0.2588
Italian	0.3641	0.1112	0.0265	0.0035	0.0384	0.0047	0.2209

Table 6.3: Multilingual Date Model: summarization and date selection scores

With the addition of the new dates to the set of available dates for the date selection phase, the resources are extended by including also sentences that are published on, or referring to these new dates. Therefore, a change in the summarization phase can be seen by taking a look at the summarization scores.

The additional resources can bring an additional confusion to some languages, as it is the case with the Spanish and the French language, which can be seen by taking a look at the decrease of their summarization scores using the *concat* metric. This is due to the fact that the *concat* metric calculates the overlapping

of the content between the summary and the ground-truth timeline, so when an additional content is added to the summary the percentage of overlap decreases. However, the scores of the other metrics, *agree* and *align*, are almost unaffected by the addition of resources during this approach.

For the Italian language on the other hand, the *Multilingual Date Model* approach outperforms the *Single language* approach, both in terms of summarization and date selection scores. Given the fact that the individual Italian resources are quite fewer in number compared to the other languages, the increasing of the number of dates and date range, and therefore also the expanded resources, seems to positively affect this language and improve its scores.

6.3.3 Early Translation

For the *Early Translation* approach, the results will be explored both using only one or two additional source languages. The date selection phase uses the *Multilingual Date Model* technique, providing a wider set of dates for the selection and a broader date range.

Therefore, for the scenarios in which the language resources are enriched with the addition of the two other source languages, the number of dates available for the date selection phase is equal for all the languages. This can be seen on the last column of Table 6.4. For all the other combinations of two languages, the number of dates for the date selection phase can be seen on the appropriate column.

Approach	**Spanish+French**	**Spanish+Italian**	**French+Italian**	**All**
Libya	68	87	110	124
Egypt	112	45	111	130
Syria	116	149	122	175
Yemen	97	81	84	109

Table 6.4: Early Translation: number of dates for the date selection phase

The summarization and date selection scores, averaged on all topics, can be seen in Table 6.5, where the column *Src.* refers to the source language used to enrich the target language resources, which can be Spanish (*es*), French (*fr*), Italian (*it*), or a combination of two of them.

Taking a look at the Spanish language scores, we can see that the Spanish language shows quite lower summarization scores for all the metrics, both compared

Lang.	Src.	concat		agree		align+m:1		Date sel.
		R1	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Spanish	fr	0.2609	0.1167	0.0683	0.0301	0.0902	0.0336	0.2697
	it	0.2375	0.1065	0.0506	0.0240	0.0688	0.0265	0.2158
	fr+it	0.2111	0.1011	0.0517	0.0226	0.0711	0.0251	0.2305
French	es	0.4333	0.1600	0.0560	0.0108	0.0889	0.0156	0.2731
	it	0.3580	0.1328	0.0464	0.0093	0.0778	0.0135	0.2310
	es+it	0.3249	0.1281	0.0467	0.0089	0.0758	0.0127	0.2588
Italian	es	0.4209	0.1382	0.0376	0.0049	0.0542	0.0066	0.2164
	fr	0.4037	0.1304	0.0405	0.0053	0.0607	0.0071	0.2048
	es+fr	0.4073	0.1406	0.0418	0.0057	0.0593	0.0074	0.2209

Table 6.5: Early Translation: summarization and date selection scores

to the *Single Language* and the *Multilingual Date Model* approaches. The only exception to this is for the *align* metric when the translated French resources are used as source language, resulting with quite similar performance between all the approaches, so we can understand that the Spanish resources are positively affected by the addition of these translated resources.

Considering the date selection scores, the combination of Spanish with the translated French resources increase the F_1 score slightly, outperforming the previous approaches. The date selection scores for the other combinations are unchanged or slightly lower when compared to the other approaches.

The French language scores show that this language is positively affected by the additional translated resources from the Spanish language, resulting with a significant increase in the date selection score when compared to the both previous approaches. Regarding the summarization ROUGE scores, they are on average slightly lower when compared to the *Single Language* and *Multilingual Date Model* approaches.
The Italian resources don't add up any benefit to the French language, which can be seen by the significant decrease of the summarization and date selection scores for this approach.

Both the Spanish and French language perform the worst in the scenario when the remaining two languages are used as source, resulting with a notable decrease in the summarization scores for all the metics. The date selection scores, when the combination of the three languages is used, are of course equal to the ones during the previously described approach, *Multilingual Date Model*, given that the same date selection technique is used for both of them.

The Italian language is the only one that has proved to be positively affected by the addition of other language resources, outperforming the summarization scores of both the *Single language* and *Multilingual Date Model* approaches, for all the metrics.
The date selection scores when one source language is added are slightly lower than the *Multilingual Date Model* approach ones, although they outperform the scores during the *Single language* approach.

Given the fact that it is the language with the lowest resources, Italian language has proved to benefit from the additional knowledge brought by the extra language resources. It has a positive influence even from the addition of the other two languages, which is not the case with the previous languages.

6.3.4 Mid Translation

Table 6.6 presents the results of the *Mid Translation* approach, averaged on all the topics, for the languages of interest. The second column *Further summ.* refers to whether the further summarization has been applied at the end of the process or not.

	Further	**concat**		**agree**		**align+m:1**		**Date sel.**
Lang.	**summ.**	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Spanish	no	0.1186	0.0635	0.0319	0.0149	0.0464	0.0171	0.2305
	yes	0.2288	0.1001	0.0519	0.0203	0.0742	0.0232	
French	no	0.2373	0.1009	0.0359	0.0061	0.0599	0.0094	0.2588
	yes	0.3335	0.1278	0.0440	0.0067	0.0745	0.0106	
Italian	no	0.3919	0.1358	0.0380	0.0047	0.0543	0.0062	0.2209
	yes	0.4015	0.1379	0.0390	0.0051	0.0564	0.0066	

Table 6.6: Mid Translation: summarization and date selection scores

The use of a further summarization step has proved to be more successful considering all the metrics, as it can be seen comparing the results of the two techniques in Table 6.6. This is related to the fact that without performing the further summarization, the summary length is quite longer than the preferred length (parameter #avg_sents), and there is a higher possibility that the summary contains more "noise" i.e. unnecessary information, thus decreasing the content overlap.
Performing this further summarization step limits the summary length and removes the less relevant information from its body, resulting with significantly higher

summarization scores for all the metrics.

Comparing the summarization scores (with the use of further summarization) during this approach, to the ones during the *Early Translation's* scenario where the combination of the three languages is considered, we can notice that there isn't a big difference between them. This behaviour is expected since the source resources (translated resources from all three languages) to be used are the same, just with a small change in the order of the steps during the process.

However, when compared to the *Early Translation* scenarios in which only one additional language was used as a source, the *Mid Translation* scores are on average slightly lower.

If we compare these results, with the ones of the *Single Language* and *Universal Date Translation*, we can conclude that the *Mid Translation* approach has the lowest summarization scores, apart from the case for the Italian language which outpeforms the previosuly mentioned approaches across all the summarization metrcis.

6.3.5 Late Translation

As mentioned previously, three different summarization methods have been tested for the *Late Translation* approach, producing almost the same results, with a slightly higher performance for the Submodular method. Thus, this summarization method has been chosen for the presentaton of the experimental results in this section.

Since the candidate sentences for the summarization phase contain resources in different language, the need for a multilingual Deep Learning model occurs for the representation of sentence embeddings. The two models that have been tested are the *SBERT* and *FastText* multilingual models, and the results of both of them will be compared.

As explained in the previous chapter, there are two possibilities for the sentences to be included for the final summary. The first one is including only the sentences that are in the target language for the construction of the final summary (referred to as *tgt*), while the latter one includes all the candidate sentences, after first translating them to the target language (referred to as *all*).

Table 6.7 shows the comparison between these two scenarios for the sentences in the final summary, using the *SBERT* model for the sentence embeddings and

the *Submodular* method for the summarization. The results are shown separately for each language, and are averaged on all topics. The column *Sents.* specifies the sentences that have been used for the construction of the final summary.

		concat		**agree**		**align+m:1**		**Date sel.**
Lang.	**Sents.**	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Spanish	tgt	0.3890	0.1541	0.0687	0.0360	0.0865	0.0384	0.2291
	all	0.2418	0.1060	0.0523	0.0204	0.0759	0.0235	
French	tgt	0.4869	0.1612	0.0514	0.0080	0.0829	0.0131	0.2377
	all	0.3547	0.1368	0.0462	0.0077	0.0790	0.0124	
Italian	tgt	0.3430	0.1005	0.0163	0.0014	0.0251	0.0021	0.2208
	all	0.4014	0.1407	0.0426	0.0064	0.0603	0.0079	

Table 6.7: Late Translation: SBERT, summarization and date selection scores

In general, the use of only the sentences which are in the target language from the pool of candidate sentences (*tgt*), appears superior in comparison to the one when all the translated sentences are used (*all*).

The only exception of this trend is the Italian language which performs better when the resources of all the languages are used for the final summary. In this scenario, when all the sentences have been used, the *Late Translation* approach scores outperform all the previous approaches for the Italian language, having slightly higher scores than the other approaches when the translation technique has been used (*Early* and *Mid Translation*), and significantly higher scores when compared to the *Multilingual Date Model* and *Single language* approaches.

The Spanish and French language on the other hand, have higher results in the scenario when only the target language sentences have been used for the final summary. In this scenario (*tgt*) they show far better results and outperform all the previous approaches, including even the *Single language* approach.

Table 6.8 shows the results of the *Late Translation* approach using the *FastText* model for the representation of the sentence embeddings.

Also for the *FastText* model, we can notice the trend of better performance when only the target language sentences are selected for the construction of the final summary, with the exception of the Italian language, which, as before, results with higher scores in the scenario when all the sentences are used.

For the Spanish language the summarization scores don't differ a lot between the two models (*SBERT* and *FastText*) for the scenario where all the sentences are

Lang.	Sents.	concat		agree		align+m:1		Date sel.
		R1	**R2**	**R1**	**R2**	**R1**	**R2**	F_1
Spanish	tgt	0.3674	0.1369	0.0709	0.0363	0.0903	0.0382	0.2291
	all	0.2457	0.1016	0.0505	0.0195	0.0730	0.0218	
French	tgt	0.4902	0.1494	0.0439	0.0068	0.0709	0.0107	0.2377
	all	0.3611	0.1341	0.0464	0.0073	0.0762	0.0108	
Italian	tgt	0.3336	0.0943	0.0152	0.0012	0.0250	0.0022	0.2208
	all	0.3432	0.1139	0.0303	0.0037	0.0458	0.0049	

Table 6.8: Late Translation: FastText, summarization and date selection scores

used. On the other hand, taking only the target language sentences into account, the *SBERT* model shows slightly better results for the summarization phase.

The French language also has better summarization scores in the case when only the target language sentences are used, performing on average similarly as the *SBERT* model.

As it was the case using the *SBERT* model, the combination of the *Late Translation* approach with the *FastText* model outperforms all the previous approaches with far better summarization scores for the Spanish and French languages.

For both scenarios the *Multilingual Date Model* has been used during the date selection phase, although it can be seen from the tables that the date selection scores during the *Late Translation* approach are slightly lower when compared to the other approaches that use the same technique. This is due to the some errors during the language detection or translation process, so those dates wouldn't be considered for the final summary.

Chapter 7

Conclusion and Future work

The Timeline Summarization (TLS), as a process that helps following the timeline of the events of interest, is the main argument that has been covered in this thesis. With the increasing number of resources on the internet, the amount of news articles to analyze keeps growing. Given the fact that constructing these timelines manually is a long and time-consuming process, the automatic extractive text summarization has been the commonly used approach.

The main objective in this work was focused on the proposal of using multilingual resources in order to enrich a target language resources by providing additional knowledge from the other languages. A new dataset has been constructed for that purpose, containing multilingual resources, in the form of news articles, related to four topics of interest. Most of the works previously done on TLS are focused only on a monolingual scenario, and although some of them can be portable to other languages, they are still unable to combine knowledge that has been extracted from news articles in different languages.

Several methodologies have been proposed in order to reach this objective, and the obtained results of each of them have been demonstrated and compared. According to the experiments that have been carried out, it can be concluded that, on average, the use of the multilingual resources is indeed beneficial. Among the different translation techniques that were the base of the conducted experiments, the *Late Translation* one has shown to significantly improve the quality of the generated summaries.

The Italian language is the one that has been proved to show the greatest improvements using the additional knowledge provided by the other languages,

increasing it's summary quality significantly. The other languages, in some of the proposed approaches, seem to be affected negatively by the concatenation of the additional knowledge provided, resulting with a decreased summary quality. However, considering the *align* metric of the used ROUGE scores for the summarization phase, this decrease is not so significant. This metric can give us the best insight of the summary quality in the context of TLS, since it constructs an alignment between the dates of the generated summary and the ground-truth timeline.

During the scope of the experiments, different extractive text summarization techniques have been tested, depending on the language scenario in which they were used. In a scenario where the source language of the input and the target language of the generated summary are the same, the TextRank algorithm has outperformed the other summarization methods. In the cross-lingual scenario where the input is in several languages, while the summary is in a specific target language, the need for using a Deep Learning language model appeared. Among the methods that can make use of this model and summarize resources in several different languages, the implementation of the Submodular method has shown superior results.

7.1 Future work

TLS has received a lot of attention in the last decades and continues to prove useful for the construction of daily summaries for important events. The proposed usage of multilingual resources is quite beneficial given the fact that major events are usually reported in many languages, each of them portraying it from a different perspective. Although this work examined the possibility to perform the TLS process in a cross-lingual scenario, there are several points that could be done for its further development.

The data is the key to constructing a good TLS system and improve the quality of our generated summaries. Carefully selecting the data, both the input news articles and the ground-truth timelines, is an important thing to consider. For the sake of simplicity, the constructed dataset used in this work, contains a rather small number of news articles, so increasing the dataset size is one of the first aspects for improving this work.

The experiments carried out in this work are using the multilingual resources in three languages and four topics of interest. However, the list of languages, as well as topics, could be further extended in order to have better coverage of even more news articles related to different events, from news agencies from different

countries. These extended resources could represent different economical, cultural or political reflections, since all of these different aspects should be taken into account and combining the knowledge of these additional resources can further improve the summary quality.

Lastly, another possible direction for the future work could be to explore the *abstractive* TLS methods for the contruction of the summaries. The abstractive text summarization combines the textual information from different sentences and create a summary that differs from the original text sentences. The constructed summary is more similar to the manual summary, therefore resulting with a better quality than the extractive approaches. This could be very beneficial for the proposed multilingual TLS system.

Bibliography

[1] Demian Gholipour Ghalandari and Georgiana Ifrim. *Examining the State-of-the-Art in News Timeline Summarization*. 2020 (cit. on pp. 6, 15, 31).

[2] Sebastian Martschat and Katja Markert. «A Temporally Sensitive Submodularity Framework for Timeline Summarization». In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. 2018, pp. 230–240 (cit. on pp. 6, 14, 15).

[3] Hai Leong Chieu and Yoong Keok Lee. «Query based event extraction along a timeline». In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2004, pp. 425–432 (cit. on pp. 6, 14).

[4] Mohammad Alrifai Giang Binh Tran and Dat Quoc Nguyen. «Predicting Relevant News Events for Timeline Summaries». In: *The 22nd International Conference on World Wide Web*. 2013, pp. 91–92 (cit. on pp. 8, 13, 47).

[5] Mohammad Alrifai Giang Tran and Eelco Herder. «Timeline Summarization from Relevant Headlines». In: *Proceedings of the 37th European Conference on Information Retrieval*. 2015, pp. 245–256 (cit. on pp. 8, 13).

[6] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Tech. rep. 1999 (cit. on pp. 9, 30).

[7] Rada Mihalcea and Paul Tarau. «Textrank: Bringing order into text». In: *Proceedings of the 2004 conference on empirical methods in natural language processing*. 2004, pp. 404–411 (cit. on pp. 10, 33).

[8] J. M. Kleinberg. «Authoritative sources in a hyperlinked environment». In: *J. ACM* 46 (1999), pp. 604–632 (cit. on pp. 10, 30).

[9] Herings, P.J.J., van der Laan, G., and D. Talman. «Measuring the power of nodes in digraphs». In: *Research Memorandum 007, Maastricht University*. 2011 (cit. on p. 10).

[10] Dragomir R. Radev, Hongyan Jing, Małgorzata Stys, and Daniel Tam. «Centroid-based summarization of multiple documents». In: *Information Processing /& Management* 40.6 (2004), p. 2004 (cit. on pp. 10, 35, 43).

[11] Demian Gholipour Ghalandari. «Revisiting the Centroid-based Method: A Strong Baseline for Multi-Document Summarization». In: *In Proceedings of the Workshop on New Frontiers in Summarization* (2017), pp. 85–90 (cit. on pp. 11, 35, 43).

[12] Elena Baralis, Luca Cagliero, Saima Jabeen, and Alessandro Fiori. «Multi-document summarization exploiting frequent itemsets». In: *Proceedings of the ACM Symposium on Applied Computing* (2012), pp. 782–786 (cit. on p. 11).

[13] Hui Lin and Jeff Bilmes. «A class of submodular functions for document summarization». In: *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics* (2011), pp. 510–520 (cit. on pp. 11, 35, 43).

[14] Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. «Summarization Based on Embedding Distributions». In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1984–1989 (cit. on p. 11).

[15] L. Chen and M. L. Nguyen. «Sentence Selective Neural Extractive Summarization with Reinforcement Learning». In: *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*. 2019, pp. 1–5 (cit. on p. 12).

[16] Cagliero Luca La Quatra Moreno. «End-to-end Training For Financial Report Summarization». In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation* (2020), pp. 118–123 (cit. on p. 12).

[17] Russell Swan and James Allan. «Automatic Generation of Overview Timelines». In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2000), pp. 49–56 (cit. on p. 12).

[18] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. «Evolutionary timeline summarization: a balanced optimization framework via iterative substitution». In: (2011), pp. 745–754 (cit. on p. 12).

[19] Remy Kessler, Xavier Tannier, Caroline Hagege, Veronique Moriceau, and Andre Bittar. «Finding salient dates for building thematic timelines». In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*. 2012, pp. 730–739 (cit. on p. 13).

[20] Eelco Herder Giang Tran and Katja Markert. «Joint graphical models for date selection in timeline summarization». In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 2015, pp. 1598–1607 (cit. on p. 13).

[21] Julius Steen and Katja Markert. «Abstractive Timeline Summarization». In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. 2019, pp. 21–31 (cit. on p. 13).

[22] Lu Wang, Claire Cardie, and Galen Marchetti. «Socially-informed timeline generation for complex events». In: *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2015), pp. 1055–1065 (cit. on pp. 14, 47).

[23] William Yang Wang, Yashar Mehdad, Dragomir R. Radev, and Amanda Stent. «A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization». In: *In Proceedings of the 22nd Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2016), pp. 58–68 (cit. on pp. 14, 47).

[24] Kiem-Hieu Nguyen, Xavier Tannier, and Veronique Moriceau. «Ranking Multidocument Event Descriptions for Building Thematic Timelines». In: (2014), pp. 1208–1217 (cit. on p. 14).

[25] Jiwei Li and Claire Cardie. «Timeline Generation: Tracking individuals on Twitter». In: (2014) (cit. on p. 14).

[26] Z. Wang, L. Shou, K. Chen, G. Chen, and S. Mehrotra. «On Summarization and Timeline Generation for Evolutionary Tweet Streams». In: *IEEE Transactions on Knowledge and Data Engineering* (2015), pp. 1301–1315 (cit. on p. 14).

[27] I. Dan Melamed. «Automatic Evaluation and Uniform Filter Cascades for Inducing N-Best Translation Lexicons». In: *Third Workshop on Very Large Corpora*. 1995 (cit. on p. 16).

[28] Martin Rajman and Tony Hartley. «Automatically predicting MT systems rankings compatible with Fluency, Adequacy or Informativeness scores». In: (2001) (cit. on p. 16).

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. «Bleu: a Method for Automatic Evaluation of Machine Translation». In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 2002, pp. 311–318 (cit. on p. 16).

[30] Chin-Yew Lin. «ROUGE: A package for automatic evaluation of summaries». In: *In Proceedings of the Text Summarization Branches Out Workshop at ACL '04* (2004), pp. 74–81 (cit. on p. 16).

[31] Sebastian Martschat and Katja Markert. «Improving rouge for timeline summarization». In: *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Vol. 2* (2017), pp. 285–290 (cit. on pp. 17, 46, 48).

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. «Efficient Estimation of Word Representations in Vector Space». In: (2013) (cit. on p. 17).

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. «Attention Is All You Need». In: (2017) (cit. on p. 18).

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 1 (Long and Short Papers) (2019), pp. 4171–4186 (cit. on p. 18).

[35] Yang Liu and Mirella Lapata. «Text Summarization with Pretrained Encoders». In: (2019) (cit. on p. 18).

[36] Derek Miller. «Leveraging BERT for Extractive Text Summarization on Lectures». In: (2019) (cit. on pp. 18, 34).

[37] Elvys Linhares Pontes, Stéphane Huet, Juan-Manuel Torres-Moreno, and Andréa Carneiro Linhares. In: *Cross-Language Text Summarization using Sentence and Multi-Sentence Compression 23rd International Conference on Natural Language /& Information Systems (NLDB)*. 2018, pp. 467–479 (cit. on p. 19).

[38] Nils Reimers and Iryna Gurevych. «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks». In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 2019, pp. 3982–3992 (cit. on p. 19).

[39] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. «Enriching Word Vectors with Subword Information». In: *In Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146 (cit. on p. 19).

[40] Soheil Danesh, Tamara Sumner, and James H. Martin. «SGRank: Combining Statistical and Graphical Methods to Improve the State of the Art in Unsupervised Keyphrase Extraction». In: *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*. 2015, pp. 117–126 (cit. on p. 22).

[41] Gertz Str"otgen. «Multilingual and Cross-domain Temporal Tagging». In: *In Language Resources and Evaluation* 47 (2013), pp. 269–298 (cit. on p. 24).

[42] Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. «Combining Graph Degeneracy and Submodularity for Unsupervised Extractive Summarization». In: *Proceedings of the Workshop on New Frontiers in Summarization*. 2017 (cit. on p. 34).

[43] Hao Zheng and Mirella Lapata. *Sentence Centrality Revisited for Unsupervised Summarization*. 2019 (cit. on p. 34).

[44] *Spacy library for NLP tasks*. `https://spacy.io/api` (cit. on p. 44).

[45] *Textacy library*. `https://pypi.org/project/textacy/0.2.3/` (cit. on p. 44).

[46] *GoogleNews library*. `https://pypi.org/project/GoogleNews/` (cit. on p. 44).

[47] *BeautifulSoup library*. `https://pypi.org/project/beautifulsoup4/` (cit. on p. 44).

[48] *tilse toolkit*. `https://github.com/smartschat/tilse/` (cit. on p. 44).

[49] *Google Translate API*. `https://pypi.org/project/google-trans-new/` (cit. on p. 44).

[50] *Sentence Transformer*. `https://pypi.org/project/sentence-transformers/` (cit. on p. 44).

[51] *FastText library*. `https://fasttext.cc/` (cit. on p. 44).

[52] *NetworkX package*. `https://networkx.org/` (cit. on p. 44).

[53] *Scikit-learn machine learning library*. `https://scikit-learn.org/stable/` (cit. on p. 44).

[54] *Numpy library*. `https://numpy.org/` (cit. on p. 45).

[55] *Gensim library*. `https://pypi.org/project/gensim/` (cit. on p. 45).

[56] Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. «Summarizing a document stream». In: *Proceedings of the 33rd European Conference on Information Retrieval*. 2011, pp. 177–188 (cit. on p. 47).

[57] Rui Yan, Liang Kong, Congrui Huang, Xiajun Wan, Xiaoming Li, and Yan Zhang. «Timeline generation through evolutionary trans-temporal summarization». In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 2011, pp. 433–443 (cit. on p. 47).

[58] Kiem-Hieu Nguyen, Xavier Tannier, and Veronique Moriceau. «Ranking multi document event descriptions for building thematic timelines». In: *Proceedings of the 25th International Conference on Computational Linguistics*. 2014, pp. 208–1217 (cit. on p. 47).

[59] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M.: Okapi at Trec-3 Gatford. «In: Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4». In: 1994 (1994), pp. 109–126 (cit. on p. 52).