TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING

DIPLOMA THESIS

Multi-Document Text Summarization

Author:
Kritharakis Emmanouil

A thesis submitted in partial fulfillment of the requirements for the master degree of Diploma in Electrical and Computer Engineering.

October 10, 2019

Abstract

In the past few years, automatic text summarization has witnessed increasing interest, since it can aid many applications by condensing the large quantities of information available into short, concise summaries. In this direction, text summarization with sequence-to-sequence (seq2seq) models has attracted the interest of the research community. Similar encoder-decoder architectures have also been exploited on multi-document text summarization. However, the adaptation of the seq2seq models to the multi-document summarization task is not always successful and requires advanced attention mechanisms to avoid unnecessary repetitions. In this thesis, we propose a novel attention mechanism, which is based on sentence similarity, to improve the multi-document text summarization process. With the proposed attention mechanism, the text summarizer takes into account the semantic and syntactic nature of the sentences, which is particularly useful in a multi-document dataset. o investigate the effectiveness of the sentence similarity algorithm, two families of experiments were conducted. In the first, we compared the proposed algorithm to a similar, recently published, sentence similarity method. Using the Pearson correlation coefficient and other statistical metrics, we prove that our algorithm is able to obtain significantly improved performance. In the second family of experiments, we integrated the sentence similarity algorithm as an attention mechanism into the text summarizer. The evaluation of the performance under several automated metrics shows that the proposed methodology outperforms other state-of-the-art text summarization techniques on the multi-document newswire topics from the DUC-2004 and TAC-2011 datasets.

Περίληψη

Τα τελευταία χρόνια, έχει παρατηρηθεί έντονο ενδιαφέρον για την αυτόματη περίληψη κειμένων, καθώς υπάρχουν πολλές εφαρμογές που απαιτούν την συμπίεση της μεγάλης ποσότητας πληροφορίας που είναι διαθέσιμη σε μικρές, συνοπτικές περιλήψεις. Σε αυτή την κατεύθυνση, η περίληψη κειμένου με μοντέλα sequence-to-sequence (seq2seq) έχει προσελκύσει το ενδιαφέρον της επιστημονικής κοινότητας. Παρόμοιες αρχιτεκτονικές κωδικοποίησης-αποκωδικοποίησης (encoder-decoder) έχουν επίσης χρησιμοποιηθεί και σε περίληψη πολλαπλών κειμένων. Ωστόσο, η προσαρμογή των μοντέλων seq2seq πάνω σε περιλήψεις πολλαπλών κειμένων δεν είναι πάντα επιτυχής και απαιτεί εξειδικευμένους μηχανισμούς εστίασης προσοχής (attention) για την αποφυγή περιττών νοηματικών επαναλήψεων. Σε αυτή την εργασία, προτείνουμε έναν καινοτόμο μηχανισμό εστίασης προσοχής, ο οποίος βασίζεται πάνω στην ομοιότητα των προτάσεων, προκειμένου να βελτιώσουμε την περίληψη σε πολλαπλά κείμενα. Με τον προτεινόμενο μηχανισμό, το σύστημα λαμβάνει υπ' όψιν του την σημασιολογική και συντακτική φύση των προτάσεων, κάτι πολύ χρήσιμο σε δεδομένα από πολλαπλά κείμενα. Για να διερευνήσουμε την αποτελεσματικότητα του αλγορίθμου ομοιότητας των προτάσεων, διεξήγαμε δύο ομάδες πειραμάτων. Στην πρώτη, ο προτεινόμενος αλγόριθμος συγκρίνεται με έναν πρόσφατα δημοσιευμένο αλγόριθμο ομοιότητας προτάσεων. Χρησιμοποιώντας ως κριτήριο τον συντελεστή συσχέτισης Pearson και άλλες στατιστικές μετρήσεις παρατηρήσαμε ότι ο αλγόριθμος μας πετυχαίνει καλύτερα αποτελέσματα. Στη δεύτερη οικογένεια πειραμάτων, ο προτεινόμενος αλγόριθμος ενσωματώθηκε ως μηχανισμός εστίασης προσοχής σε μοντέλα seq2seq για την περίληψη πολλαπλών κειμένων. Η αποτίμηση της επίδοσης με αυτοματοποιημένες μετρικές απέδειξε ότι το προτεινόμενο σύστημα υπερβαίνει συστηματικά σε επίδοση άλλες μεθόδους που βρίσκονται στην αιχμή της τεχνολογίας (state-of-the-art) πάνω στις βάσεις δεδομένων πολλαπλών ειδησεογραφικών κειμένων DUC-2004 και TAC-2011.

Acknowledgements

First and foremost, I would like to thank Dr.Vasileios Diakoloukas, for guiding me and providing consultation at every step along in the road, for challenging me to experiment and research new fields and for the good word when needed. I would also like to thank my advisor Dr Michail G. Lagoudakis, I couldn't have done it without you. Lastly, I would like to thank my parents Foteini and John as well as my brother, Polinikis for always believing in me.

Contents

Chapter 1

Introduction

In the modern era of big data, acquiring information from a vast pool of textual documents is highly demanding. A well-known example is the social media networks. In 2019, every second thousand of new textual messages, posts, and articles are uploaded and shared through all the various networks [1]. Modern users indeed look more and more for concise information, and they want it faster than ever. Therefore there is an increasing need for automatic systems capable of shortening a text document into a text form that contains the primary information from the original.

1.1 Motivation

Automated summarization provides a proper solution to this need. With a shorter version of the textual documents, the text content can be acquired, processed, and digested effectively and efficiently. Nevertheless, contrary to other successful tasks in the Natural Language Processing (NLP) like machine translation, where many solutions are commercially published since many years [2], automatic text summarization systems are still a bit far from producing business-reliable outputs, despite the fact that automatic text summarization is not a new research area with the first scientific results being published in 1958 [3]. The success of neural networks through the years gave extra motivation for the research community to work on a multi-document text summarization field in order to achieve similar progress. Although existing single-document text summarization approaches achieve to produce short, concise summaries [52], multi-document text summarization works are not able to produce meaningful summaries. The root of this problem is that the majority of multi-document text summarization approaches do not take into consideration the semantic and syntactic nature of the input document sentences. This problem motivated us to find a solution and proposed it in our thesis.

1.2 Thesis Contributions

The thesis aims to study the possibility of developing a new sentence similarity algorithm able to recognize the semantic and syntactic nature of sentences in order to improve a state-of-the-art multi-document text summarization model [49]. This multi-document model contains a Maximal Marginal Relevance (MMR) algorithm

Figure 1.1: The State-of-the-art multi-document text summarization model with our proposed sentence similarity algorithm

and a single document text summarization model. The MMR algorithm is responsible for finding the most representative sentences from all the multiple documents and input them into the single document model, which is going to output the final summary. The degree of representation is assigned to an MMR score. This score is calculated based on how important are the sentences related to all the multiple documents and how redundant they are based on the partial generated summary. In this work, we propose an alternative way to calculate the redundancy part of the representative sentences based on a sentence-similarity algorithm. The more semantically and syntactically related the document sentences with the partial generated summary are, the more redundant they are to produce the next sentence of the summary. Based on this approach, we succeed in producing meaningful summaries from multiple documents. Figure 1.1 presents our contribution in the state-of-the-art multi-document model, which is the incorporated sentence similarity algorithm in the classical MMR algorithm.

To investigate the effectiveness of the sentence similarity algorithm, two families of experiments were conducted. In the first, we show that our proposed sentence similarity algorithm outperforms a similar, recently published, sentence similarity method [24] based on statistical measurements. In the second family of experiments, we compare the state-of-the-art multi-document model with our new, improved multi-document model using two multi-document datasets DUC-04 and TAC-11, and we show that our work has better results than the state-of-the-art model based on several automatic evaluation metrics.

1.3 Thesis Outline

In the next chapter, we present an overview of the related work on the field of automatic multi-document text summarization and give the outcome of the background research that was carried in the various topics required to achieve the thesis goal, such as natural language processing techniques and information on deep neural networks. After that, we go through a review of the single document model and its mechanisms, which are used as a component in the state-of-the-art multi-document model. In chapter 4, we describe all the components of our sentence similarity algorithm, and we cite examples to illustrate how does it work. In chapter 5, we present the state-of-the-art multi-document model and our proposed sentence similarity algorithm implemented on it. Through chapter 6, both text summarization and sentence similarity experiments are carried out, including information about their evaluation metrics used to compare our proposed sentence-similarity algorithm firstly with a recently published one and secondly the new multi-document text summarization model with the baseline model and other multi-document techniques. The last chapter concludes the thesis and explores the future work of the existent system.

Chapter 2

Background And Related Work

In this chapter, we present an overview of the various text summarization categories and a wide range of its applications. Then we go through several natural language processing techniques that will be used in later chapters, before exploring the sequence to sequence learning using neural networks. Moreover, we give a detailed overview of all the evaluation metrics we utilize in our experiments, and last, but not least significant related works in the field of single and multiple text summarization fields will be presented.

2.1 Auto Text Summarization

2.1.1 Genres Of Summarization

Text summarization is a complex and multifarious domain in the field of Natural Language Processing (NLP). For that reason, there are several genres of summarization based on number of documents to summarize, type of summary or even type of input and output language. Below we present these there genres of text summarization:

Summarization categories based on type of generated summary:

> **Extractive Summarization** : A summarization is considered to be *extractive* if it extracts selected passages from the source text, then arranges them to form a summary. The vast majority of existing approaches to automatic summarization are extractive, mostly because it is much easier to select text than it is to generate text from scratch. To elaborate it, if the extractive approach involves selecting and rearranging whole sentences from the source text, it is guaranteed that the final product is grammatically correct, reasonably readable, and related to the source text. These systems (several are available online) can be reasonably successful when applied to mid-length factual texts such as news articles and technical documents. On the other hand, the extractive approach is too restrictive to produce human-like summaries, especially of longer, more complex texts [27], [28], [29] .
>
> **Abstractive Summarization** : A summarization is considered to be *abstractive* if it it is capable of generating new sentences using language generation models grounded on representations of source documents. Therefore, they

have a strong potential to produce high-quality summaries that are verbally innovative and can also easily incorporate external knowledge [11],[13].

Summarization categories based on number of documents to summarize :

Single-Document Summarization : Single-DocumentSummarization was the rst approaches towards developing an automated summarization system. Here, the system takes one document as an input and produces a concise summary of the input document. Research community has shown success works through the years especially on the sequence to sequence architecture family [13], [52].

Multi-Document Summarization : Based on the high demanding need to compress textual information from multiple resources to short text, researchers interests shifted towards the problem of multi-document summarization. Here, the system takes multiple documents referring to same topic and produces a single summary of the original multi-document articles. These systems follows two fold approaches: (a) Sentence Ranking - sentence score is related on how informative it is [30]. (b) Sentence Selection - selecting top k sentences based on ranking score such that the sentences selected in the summary should not be redundant [49].

Summarization categories based on type of input and output language :

Monolingual Summarization : A Summarization is considered to be Monolingual if the input and output language in the model is the same [13], [49].

Multilingual Summarization : In this category the input and output language is again the same but the model can work e ciently with multiple languages [31].

Cross-lingual Summarization : Living in a multicultural world, the increasing need to shorten textual documents from one language to another lead to this category [32].

2.1.2 Applications

In the past few years, text summarization has been used in a wide variety of applications. First, summarizing product reviews is one of the most common applications since customers are able to receive concise information about a product in a short amount of time [4]. Another one was useful for educational purposes, such as student responses to post-class questionnaires since the traditional way includes instructors, who manually analyze these responses in a costly manner [5]. Moreover, spreading the news in a shorter version is essential nowadays, and researchers tried to summarize sets of news articles discussing speci c topics into short texts [6]. An uncommon eld where text summarization plays an important role is the nancial research. brokers read every day a vast amount of information related the behavior of common stocks and a summarization of all this information saves them a lot of money [38]. Last, but not least, eld is the biomedical one, in which text summarization plays an important role. In particular, Clinical summarization can be de ned as the act of collecting, distilling, and synthesizing patient information to facilitate any of a wide range of clinical tasks. Examples of high-level summarization, such as the discharge summary, daily progress notes, patient hando at a change of shift, and oral case presentation, are commonplace in medicine [7] .

2.2 Natural Language Processing Techniques

2.2.1 Text Wrangling Methods

Working with textual data requires much prepossessing work in order to transform and map data from their \raw" form into another more e cient format. This format allows us to extract all the informational content of the textual data. Text wrangling methods such as tokenization, special character and stopwords removal, lemmatization, and stemming are responsible for the preprocessing work. In the text summarization eld, the text wrangling methods play an essential role in order to extract most representative sentence's words and eliminate words with limited information content. Below we present some of these methods.

Tokenization : Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols, and other elements called tokens. A token is the smallest text unit a machine can process. Sometimes, it makes sense for the smallest unit to be either a word or a letter. In the process of tokenization, some characters like punctuation marks are discarded. For instance, in a summarization model if the input sentence is \Today is a sunny day.", the process of tokenization will break the sequence in tokens \Today", \is", \a", \sunny", \day" and \.".

Special Character and Stopwords Removal : Special characters are usually non-alphanumeric characters or even occasionally numeric characters, which add to the extra noise in textual input. Words that have little or no signi cance, especially when constructing meaningful features from the text, are known as stopwords. These are usually words that end up having the maximum frequency in a corpus. Typically, these can be articles, conjunctions, prepositions. Both special characters and stopwords should be removed from the input textual data in the automatic summarization in order to increase input's informational content.

Stemming : Word stems are also known as the base form of a word since new words can be created by attaching a xes to them in a process known as in ection. For example, adding a xes to the word \jump" new words like \jumps", \jumped", or \jumping" can be formed. In this case, the base word \jump" is the word stem. The reverse process of obtaining the base form of a word from its in ected form is known as stemming. Stemming helps in standardizing words to their base or root stem, irrespective of their in ections, which is essential in information retrieval of the summarization process.

Lemmatization : Lemmatization is very similar to stemming where word a xes are removed to get to the base form of a word. However, the base form in this case is known as the root word, but not the root stem. The di erence being that the root word is always a lexicographically correct word (present in the dictionary), but the root stem may not be so. Thus, root word, also known as the lemma, will always be present in the dictionary. For example, the lemma of word “are” is the word “be”.

2.2.2 Word Embeddings

In order to model word sequences with mathematical models such as the seq2seq neural network architectures, commonly used in the automatic text summarization process, it is necessary to obtain a representation of the words into a numerical encoding. This numerical encoding is often called word embeddings. There are several such representations that could be used. Among the most popular ones in the context of NLP is the one hot-word encoding.

Historically, words, in the context of natural language processing, are encoded as one-hot encoding. This method requires a vocabulary-sized vector where all-but-one of its elements are 0, and a single element, the word index, is 1. For example, if the index corresponding to the word “text” is 32, the corresponding vector is $[x_0 = 0; \ldots; x_{32} = 1 \ldots; x_{M-1} = 0]$ where M is the vocabulary size. Similarly, assuming the index corresponding to the word “document” is 1821, we receive the following vector $[x_0 = 0; \ldots; x_{1821} = 1 \ldots; x_{M-1} = 0]$.It is worth mentioning that the corresponding vector representations which are usually noted as word embeddings, are orthogonal. A big disadvantage of this representation is that possible semantic similarity that might exist between the words, is not considered; their vector representation is completely independent. This means that the word “dog” could have the same distance as the word “cat” but also with the word “sun”. An additional issue with the one-hot representation of words is the size of the vectors, as each vector is the size of the vocabulary while only representing a single index. For these reasons, alternative methods are utilized nowadays in the field of machine learning like Word2vec.

Word2vec is the most common representation that aims to capture the semantic understanding of a word in a dense continuous smaller dimensional space. It was developed by Mikolov [20]. Mikolov and his team suggested two unsupervised methods for better representation, the Continuous bag-of-words (CBOW), and the Skip-gram methods. These methods are unsupervised, as there is no true observed representation for words. In particular, on the CBOW method users try to train a representation for a given window of words of size 2k, $[w_{k-1}; \ldots; w_{i-1}; w_{i+1}; \ldots; w_{k+1}]$ in order to predict the central word w_i. On the other hand, on the skip-gram method, the opposite procedure is followed since for a given word w_i users should predict the 2k size window of words. In [20], the authors developed an efficient algorithm, including Hierarchical Softmax, Negative Sampling, and Sub-sampling of Frequent Words to train those unsupervised methods over a huge corpus (around 100 billion words).

In many neural models of sequence to sequence text summarization, words are encoded using pre-trained word2vec representations on a large corpus. Alternatively, for some tasks, it is efficient to train the word embeddings as part of the end to end task optimization. By this process, the word embeddings vectors are initially assigned to random values and as the model starts its training phase, the word embeddings vectors shape their final values. The latter method is the one it was used in the single-document text summarization model.

2.2.3 TFIDF Vectorization

TFIDF vectorizers are an essential tool in the Natural Language Processing field and especially in the text summarization domain. This tool is frequently used to extract a few sentences that best summarized one or multiple documents [49]. \TFIDF" stands for \Term Frequency Inverse Document Frequency". In the *Term Frequency* part, a counter *tf* measures how many times a word occurs in a given document (synonymous with bag of words). In the *Inverse Document Frequency*, is measured the number of times a word occurs in a corpus of documents. The final tf-idf value is the multiplication between these two terms. The tf-idf value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general [41]. There are many ways to express the TFIDF value of a document's word but the most common among all is given in the following formula, where w_i stands for word i, f_{wi} for frequency count of word i, d for document where word i is included and D for the number of documents.

$$w_i = f_{wi} \log(\frac{d}{D})$$

In our work, TFIDF vectorizers are essential for the importance model \Cosine" of the multiple document text summarization model [49]. The \Cosine" model calculates the importance score of document sentences based on their TFIDF vectors. Each document will be described as a unique TFIDF vector. The same procedure will be followed for every document's sentence. Applying the cosine similarity function in the document's TFIDF vector and one of its sentence's TFIDF vector, a score will be assigned in this sentence as an indicator of its importance. The sentences with the highest score are the ones with the highest informational content.

2.3 Neural Networks

2.3.1 Recurrent Neural Networks (RNNs)

In the machine learning field, researchers have to deal with different kinds of inputs for their models like numbers, words, or even sequences of them. In order to handle a sequential input, a specific type of neural network has been utilized, and it is named the Recurrent Neural Network (RNN). RNNs manage to handle the sequential nature of language by holding a state which represents all previous information passed beforehand and iteratively proceeds to the next part of the sequence, as words are read one by one, left to right. The basic component of the RNN is called the RNN cell, and its original implementation is showed with the following equation for the basic RNN cell: $h_t = \tanh(w_{ih} x_t + w_{hh} h_{t-1} + b)$, where $w_{ih}; w_{hh}; b$ are learnable parameters, h_t represent the hidden state at time t and x_t is the input at time t.

Various design patterns for recurrent neural networks have been utilized through the years for the sake of different Natural Processing Language domains, including text summarization. Figure 2.1 depicts all the different design patterns. Input vectors are in purple; output vectors are in pink, and yellow vectors hold the RNN's state. The most straightforward design pattern is called \One to one,", which is the vanilla mode of processing without RNN, from fixed-sized input to fixed-sized

Figure 2.1: Various RNN design patterns [23]

output, and it is commonly used in image classi cation [33]. The next one is called \one to many", in which a single input is taken, and we get a sequence output. It is widely used in Image captioning [34], where it takes an image and outputs a sentence of words. The third in the row design pattern is \many to one," in which a sequence input is given, and the result is a single output. For example, in sentiment analysis, a given sentence is classi ed as expressing positive or negative sentiment. The last two design patterns are called \many to many", in which a sequence is given as an input, and output is also a sequence. In these two patterns the signi cant di erence is that the left one produce an output, which is not symmetrical to the input like the right one. The text summarization belongs to the \many-to-many" category with the unsymmetrical output, since the summaries are mostly shorter than the original document, with a vast list of successful works through the years [52], [49].

Despite its current success, recurrent neural networks did not work that e ciently back in 1990, and the problem was the Vanishing Gradient Decent. The gradient descent algorithm nds the global minimum of the cost function that is going to be an optimal setup for the neural network. The information travels through the neural network from input neurons to the output neurons, while the error is calculated and propagated back through the network to update the weights. Nevertheless, the problem with the RNNs is that information travels through time in RNNs, which means that information from previous time points is used as input for the next time points. This is the root of the problem caused when a node tries to minimize its error every single neuron that participated in the calculation of the output, associated with this cost function, should have its weight updated. Moreover, the thing with RNNs is that it is not just the neurons directly below this output layer that contributed but all of the neurons far back in time. So, the propagation goes back through time to these neurons. However, the constant multiplication of RNN's weights(values close to zero) backward leads to vanishing the gradient. The lower the gradient is, the harder it is for the network to update the weights, and the longer it takes to get to the nal result. In 1991, Joseph Hochreiter [37] found the Vanishing gradient Decent problem and proposed a new RNN implementation in 1997 [36] called Long Term Short Memory network (LSTM), which solves this problem. In our work, we are going to use LSTMs both in the encoder and the decoder of the multi-document text summarization model.

Figure 2.2: The LSTM cell [40]

2.3.2 Long Short Term Memory Networks (LSTMs)

LSTMs cells are far more complicated than the baseline RNN cell. One of their most signi cant advantages is that they do not su er from the Vanishing Gradient Decent problem. LSTMs are explicitly designed to avoid this long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. As Figure 2.1 depicts the LSTM cell contains 3 gates (input i_t, forget f_t and output o_t), the memory cell c_t, where t represents time in order to have a better understanding about the sequence input and what information should keep or discard and the hidden state h_t, which is responsible to remember the information from all the previous t 1 states. The following equations represent the input, forget and output gates as well with the cell memory block and hidden state of time t:

$$
\begin{aligned}
i_t &= \ (W_{xi} \ x_t + W_{hi} \ h_{t\ 1} + W_{ci} \ c_{t\ 1} + b_i)\\
f_t &= \ (W_{xf} \ x_t + W_{hf} \ h_{t\ 1} + W_{fi} \ c_{t\ 1} + b_f)\\
c_t &= f_t \ c_{t\ 1} + i_t \ \tanh(W_{xc} \ x_t + W_{hc} \ h_{t\ 1} + b_c)\\
o_t &= \ (W_{xo} \ x_t + W_{ho} \ h_{t\ 1} + W_{co} \ c_{t\ 1} + b_o)\\
h_t &= o_t \ \tanh(c_t)
\end{aligned}
$$

The sets f W_{xi} ; b_i ; W_{hi} ; W_{ci} g; f W_{xf} ; b_f ; W_{hf} ; W_{fi} g; f W_{xc} ; W_{hc} ; b_c g; f W_{xo} ; W_{ho} ; w_{co} ; b_o g are learnable parameters, while represent a sigmoid function. Index t stands for time.

Based on the single LSTM network, many other alternative LSTMs have been introduced through the years. One of the most common types in text summarization is the Bidirectional Long Term Short Memory network (Bi-LSTM). This network consists of two LSTMs networks, the backward and the forward. Using bidirectional will run the input sentence in two ways, one from past to future and one from future to past and what di ers this approach from unidirectional is that in the LSTM that runs backward information from the future is preserved and using the two hidden

Figure 2.3: Bidirectional Long Short Term Memory network

states combined the preserved information comes from both past and future. Based on that, BiLSTMs show excellent results as they can understand the context better than the unidirectional [21]. The nal hidden and cell states from the network are the concatenation of the individual hidden states from the backward and forward LSTMs networks. Figure 2.3 shows a bidirectional long short term memory network. In our work, the single document text summarization model has a Bi-LSTM in the encoder in order to recognize better the sequential form of document sentences than a baseline LSTM. In the decoder phase, a baseline LSTM is utilized to produce step by step the summary words as it is described in chapter 3.

2.4 Evaluation Metrics

2.4.1 Pearson Correlation Coe cient

One of most common statistical evaluation metrics in textual semantic similarity is the Pearson correlation coe cient [24], [25]. The Pearson product-moment correlation coe cient (or Pearson correlation coe cient, for short) is a measure of the strength of a linear association between two variables and is denoted by r. Basically, a Pearson product-moment correlation attempts to draw a line of best t through the data of two variables, and the Pearson correlation coe cient, r, indicates how far away all these data points are to this line of best t. In our work, we utilize this metric to evaluate the linear association between our sentence similarity algorithm and mean human results based on 60 sentence pairs originally measure by Rubenstein and Goodenough [22].

The Pearson correlation coe cient, r, can take a range of values from +1 to -1. A value of 0 indicates that there is no association between the two variables. A value greater than 0 indicates a positive association; that is, as the value of one variable increases, so does the value of the other variable. A value less than 0 indicates a negative association; that is, as the value of one variable increases, the value of the other variable decreases. Of course, some guidelines are interpreting the Pearson's correlation coe cient and are presented in the table 2.1.

Strength of Association	Coe cient r	
	Positive	Negative
Small	0.1 to 0.3	-0.1 to -0.3
Medium	0.3 to 0.5	-0.3 to -0.5
Large	0.5 to 1	-0.5 to -1

Table 2.1: Pearson correlation coe cient association scale

2.4.2 ROUGE

In the eld of text summarization, there are two methods to evaluate a summary. Either uses an automatic evaluation metric or evaluate based on a professional linguist's opinion. Traditionally, evaluation of summarization involves human judgments of di erent quality metrics, for example, coherence, conciseness, grammar, readability, and content. On the other hand, there are plenty of automatic evaluation metrics such as BLEU [44],[45] or METEOR [46] but the most frequently used automatic evaluation metric on natural language processing eld is ROUGE [47].

The acronym ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans.

ROUGE-N: N-gram Co-Occurrence Statistics

A quick de nition about ROUGE-N is An n-gram recall between a candidate summary and a set of reference summaries[1] In particular, ROUGE-N is the percentage of all the matching words in the whole set of words in reference summaries named R as it is described in the below equation.

$$\text{ROUGE-N} = \frac{\sum_{r \in \{R\}} \sum_{word \in r} countMatch(word)}{\sum_{r \in \{R\}} \sum_{word \in s} count(word)}$$

The letter "N" describes how many consecutive words the metric is going to take into consideration for calculating the nal result. In our experiment, we used $N = 1$ and $N = 2$, which are quite common metrics on text summarization evaluation as there is a huge di erence in the percentage rate to evaluate the candidate summary. The reason behind this huge di erence is that one word ($N = 1$) is more common between a candidate summary and a set of reference summaries than a couple of consecutive words ($N = 2$).

ROUGE-L: Longest Common Sub-sequence

Before explaining the ROUGE-L evaluation metric, it is vital to de ne what is a subsequence. Based on Cormen [48]: A sequence $Z = [z_1; z_2; \ldots; z_n]$ is a sub-sequence of another sequence $X = [x_1; x_2; \ldots; x_m]$, if there exists a strict increasing sequence $[i_1; i_2; \ldots; i_k]$ of indices of X such that for all $j = 1; 2; \ldots; k$, we have $x_{i_j} = z_j$. Given two sequences X and Y, the longest common subsequence (LCS) of X and Y is a common sub-sequence with maximum length. This is exactly what ROUGE-L calculates but on a summary scale.

When applying to summary-level, we take the union LCS matches between a reference summary sentence, r_i, and every candidate summary sentence, c_j. Given a reference summary of k sentences containing a total of m words and a candidate summary of n sentences containing a total of n words, the summary-level LCS measure can be computed as follows:

$$R_{lcs} = \frac{\sum_{i=1}^{k} LCS_U(r_i;C)}{m} \quad (1)$$
$$P_{lcs} = \frac{\sum_{i=1}^{k} LCS_U(r_i;C)}{n} \quad (2)$$
$$F_{lcs} = \frac{(1+b^2)\,R_{lcs}\,P_{lcs}}{R_{lcs} + b^2 P_{lcs}} \quad (3)$$

In equation (3) is set to a very big number and $LCS_U(r_i;C)$ is the LCS score of the union longest common sub-sequence between reference sentence r_i and candidate summary C. For example, if $r_i = w_1w_2w_3w_4w_5$, and C contains two sentences: $c_1 = w_1w_2w_6w_7w_8$ and $c_2 = w_1w_3w_8w_9w_5$, then the longest common sub-sequence of r_i and c_1 is \"w_1w_2" and the longest common sub-sequence of r_i and c_2 is \"$w_1w_3w_5$". The union longest common sub-sequence of r_i, c_1; and c_2 is \"$w_1w_2w_3w_5$" and $LCS_U(r_i;C) = 4/5$:

ROUGE-S: Skip-bi gram Co-Occurrence Statistics

ROUGE-SU

Skip-bi gram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bi gram co-occurrence statistics measure the overlap of skip bi grams between a candidate translation and a set of reference translations.Using an example will be much easier to understand how ROUGE-S is working. Below we present some candidate sentences:

S_1 : Boy kissed the girl.
S_2 : Boy kiss the girl.
S_3 : The girl kiss boy.
S_4 : The girl boy kissed.

Each sentence has $C(4;2) = 6$ skip-bi grams, where $C(4;2)$ stands for the sum of combination for choosing 2 out of 4 consecutive words. For example, S_1 has the following skip-bi grams: (\"Boy kissed" , \"boy the", \"boy-girl", \"kissed the", \"kissed girl", \"the girl").

S_2 has three skip-bi gram matches with S_1 (\"boy the", \"boy-girl", \"the girl"), S_3 has one skip-bi gram match with S_1 (\"the girl"), and S_4 has two skip-bi gram matches with S_1 (\"boy kissed", \"the girl"). Given sentences X of length m and Y of length n, assuming X is a reference sentence, and Y is a candidate sentence, we compute skip-bi gram-as follows:

$$R_{skip_2} = \frac{Skip_2(X;Y)}{C(m;2)} \quad (1)$$
$$P_{skip_2} = \frac{Skip_2(X;Y)}{C(n;2)} \quad (2)$$
$$F_{skip_2} = \frac{(1+b^2)\,R_{skip_2}\,P_{skip_2}}{R_{skip_2} + b^2 P_{skip_2}} \quad (3)$$

Where $Skip_2(X,Y)$ is the number of skip-bi gram matches between X and Y, controlling the relative importance of P_{skip_2} and R_{skip_2}, and C is the combination function.

Using Equation (3) with = 1 and S_1 as the reference, S_2's ROUGE-S score is 0.5, S_3 is 0.167, and S_4 is 0.333. Therefore, S_2 is better than S_3 and S_4, and S_4 is better than S_3.

Applying skip-bigram without any constraint on the distance between the words, spurious matches such as \the the" or \of in" might be counted as valid matches. To reduce these spurious matches, we can limit the maximum skip distance, d_{skip}, between two in-order words that is allowed to form a skip-bigram. For example, if we set d_{skip} to 0 then ROUGE-S is equivalent to bigram. If we set d_{skip} to 4 then only word pairs of at most 4 words apart can form skip-bigrams. In our thesis, we set d_{skip} = 4 in order to be able to compare our results with similar techniques outcomes which used d_{skip} = 4.

ROUGE-SU: Extension of ROUGE-S

One potential problem for ROUGE-S is that it does not give any credit to a candidate sentence if the sentence does not have any word pair co-occurring with its references. For example, the following sentence has a ROUGE-S score of zero:

S_5: Girl the kissed boy.

S_5 is the exact reverse of S_1, and there is no skip bigram match between them. However, sentences similar to S_5 from sentences that do not have single word co-occurrence with S_1 should be di erent. To achieve this, we extend ROUGE-S with the addition of unigram as a counting unit. The extended version is called ROUGE-SU.

2.5 Related Work

Summarization is a function from one sequence of words to another where the input is the source document of which we want to summaries, and the output is the summary. The output should hold speci c characteristics, e.g., being shorter than the input document while preserving its salient information. Any such function is called sequence-to-sequence (or seq2seq) methods. Using seq2seq methods for text summarization purposes is a relatively new idea in the eld of machine learning. In this section, all the related work in the natural language processing eld on seq2seq text summarization techniques will be presented.

Although text summarization intrigued researchers long ago, there were only a research works that showed promising results prior to the recently developed neural network based text summarization techniques., For instance, Jing (2000) [9] presented a novel sentence reduction system for automatically removing extraneous phrases from sentences that are extracted from a document for summarization purposes. His proposed system uses multiple sources of knowledge to decide which phrases in an extracted sentence can be removed, including syntactic knowledge, context information, and statistics. Another signi cant work was the one introduced by Cheung and Penn (2014) [10]. They proposed a sentence enhancement as a novel technique for text-to-text generation in abstractive summarization. Compared to extraction of previous approaches to sentence fusion, sentence enhancement increased the range of possible summary sentences by allowing the combination of dependency subtrees from any sentence from the source text.

Back in 2015, the rst paper to use an end-to-end neural network for the abstractive summarization task was Rush [11].He was the rst to introduce a neural attention seq2seq model with an attention-based encoder and a neural network language model (NNLM) decoder to the abstractive sentence summarization task, which has achieved a signi cant performance improvement over conventional methods. Using an innovative idea like this, many scientists tried to extend it. One of them was Chopra [12], who further extended this model by replacing the feed-forward NNLM with a recurrent neural network (RNN). The model is also equipped with a convolutional attention-based encoder and in combination with the RNN decoder, outperformed other state-of-the-art models on a commonly used benchmark dataset such as Gigaword corpus.

In 2016, Nallapati [13] introduced several novel elements to the RNN encoder-decoder architecture to address critical problems in the abstractive text summarization. One of them was how to tackle out-of-vocabulary (OOV) words with a pointer mechanism. The pointer mechanism determines in the decoder whether to generate words based on the current state of the decoder or, instead, to copy words verbatim from the source. This decision was critical in the News domain typically used for summarization benchmarks because it covers the case of rare named entities, which are not processed naturally by a standard encoder-decoder architecture. Also, Nallapati tried to tackle the problem of repetitive summaries; he used a technique called coverage. The idea is that he used the attention distribution to keep track of what has been covered so far and penalize the network for attending to some parts again. Finally, in 2017, Nallapati published a neural extractive approach [15], which uses hierarchical RNNs to select sentences, and found that it signi cantly outperformed their abstractive result. This approach has the additional advantage of being very interpretable since it allows visualization of its predictions broken up by abstract features such as information content, salience, and novelty.

Another point worth mentioning in the text summarization eld was that large-scale datasets for summarization of longer text were rare. Nallapati was the rst who tried to summarize these kinds of datasets by adapting the DeepMind question-answering dataset (Hermann [14]) for summarization, resulting in the CNN/Daily Mail dataset, and provided the rst abstractive baselines.

About the multi-document text summarization, popular methods for multi-document summarization have been extractive. Meaningful sentences are extracted from a set of source documents and optionally compressed to form a summary. In 2010, Galanis [16] proposed a new method that compresses sentences by removing words. In recent years neural networks have been exploited to learn word/sentence representations for multi-document summarization. In 2017, Yasunaga [18] and his team proposed a neural multi-document summarization (MDS) system that incorporates sentence relation graphs. They employ a Graph Convolutional Network (GCN) on the relation graphs, with sentence embeddings obtained from Recurrent Neural Networks as input node features. In the same year, Cao [17] and his research team proposed a novel summarization system called TCSum, which leveraged plentiful text classi cation data to improve the performance of multi-document summarization. TCSum projects documents onto distributed representations, which act as a bridge between text classi cation and summarization. It also utilized the classi - cation results to produce summaries of di erent styles. These approaches remain extractive, and despite encouraging results, summarizing a large number of texts still requires sophisticated abstraction capabilities such as generalization, paraphrasing, and sentence fusion [19].

In this thesis, the multi-document text summarization technique is based on a single-document model [52], which has a lot in common with Nallapati's work in 2016 with minor changes in the pointer mechanism. In particular, Nallapati uses the pointer mechanism just for the out-of-vocabulary words while the single-document model we used, is allowed to freely learn when to use the pointer and mix the probabilities from the copy distribution and the vocabulary distribution. Further analytical information about the single document structure and usage for multi-document summarization will be described in the next following chapters.

Chapter 3

Single-Document Model

In this chapter, we present a state-of-the-art single document text summarization model. This model is one of the two main components of the multi-document text summarization model, so a detailed report of the baseline architecture and some important mechanisms such as the Attention, Coverage and Point Generator, which allow the model to summarize a single document efficiently, is required.

3.1 Baseline Model

The baseline sequence to sequence single document text summarization model is composed of an encoder and a decoder. The encoder is responsible to read the source article denoted by $x = [x_1; \ldots; x_J]$ (named encoding steps) and transforms it to hidden states $h^e = [h^e_1; \ldots h^e_J]$, while the decoder has to take those hidden states and produces the final summary $y = [y_1; \ldots; y_T]$ (named decoding steps). Superscripts e,d are the shortcut notations used to indicate that they are for the encoder and decoder severally. We use J and T to represent the maximum number of tokens (document length) of the source document and the summary, respectively. If the source article contains more tokens than the maximum input tokens, then it should be truncated. Both the encoder and the decoder have LSTMs architectures. However the encoder has a bidirectional LSTM (Bi-LSTM) architecture. This architecture combines a forward LSTM cell $\overrightarrow{h^e}$ with a backward LSTM cell h^e. The reason behind this choice is that bi-LSTMs check the input document from start to end and backwards, which allow it to have a greater understanding about the importance of sequential data.

For the initialization phase of the encoding stage , memory cell and hidden state will be equal to zero $c_0 = 0$ and $h_0 = 0$. Each token from the input vector $x = [x_1; \ldots; x_J]$ will be represented with its word embedding in the vector $E_x = [E_{x1}; \ldots; E_{xJ}]$, which is not pre-trained but it will be trained through the training phase of the model. Also, the encoding part will produce the hidden states as it was described in the equation 2.1 with an important change that the hidden states will be the concatenation of a hidden state from the forward and the backward part $(\overrightarrow{h^e}\ \ h^e)$.

For the initialization phase of the decoding stage, memory cell and hidden state will be equal to $h^d_0 = \tanh(W_{0d}\ (\overrightarrow{h^e_J}\ \ h^e_1) + b_{0d})$ and $c^d_0 = \overrightarrow{c^e_J}\ \ c^e_1$, where b_{0d} and W_{0d} are learnable parameters. The goal of the decoding step is to generate a summary word. In order to do that at each decoding step, we first update the hidden state h^d_t conditioned on the previous hidden states and input token. More specifically,

Figure 3.1: Single Document Text Summarization Model [53]

through a decoding step t, each LSTM cell will produce the next hidden state for the decoder $h_t^d = LSTM(h_{t\ 1}^d; E_{yt\ 1})$, where $E_{y_{t\ 1}}$ is the word embedding representation of the previous generated summary word $y_{t\ 1}$,which works as an input value to the decoder. After assigning a value to hidden state h_t^d, it is time to calculate the probability vocabulary distribution. Probability vocabulary distribution is a probability density function over all the words in the vocabulary, and it is responsible for pointing out which word ts better to the partial generated summary in order to be chosen as the next word of it. The vocabulary distribution is de ned as following : $P_{Vocab;t} = softmax(W_d \ \ h_t^d + b_d)$, where $W_d; b_d$ are learnable parameters and softmax is an activation function given by the formula $softmax(v) = \frac{P exp^v}{exp^v}$. To be graphically depicted the Figure 2.2 below presents the baseline model. As it is easily visible, the decoder has some prede ned tags for starting and ending the summary, and their names are SOS (Start of Summary) and EOS (End of Summary) respectively.

Nevertheless, it should be conceded that this baseline model su ers from a lot of technical problems. First, training through backpropagation is not e cient since encoder and decoder paths are far apart and reduce the gradient signals. Secondly, the model recognizes every input token as the same without paying attention if it is informative or not about the meaning of the sentence. Third, there is no methodology to deal with out-of-vocabulary words (OOV) and fourth consecutive repetitions over the same word as an output is a common phenomenon. The rst two problems are solved with the attention mechanism and the third with the pointer generator mechanism and the forth with the coverage mechanism which is going to present in the following sections.

3.2 Beam Search Algorithm

In order to improve the performance of seq2seq models for the task of text summarization, scientists have proposed some methods to make their summaries even better. One of them is called Beam Search Algorithm.

The single-document model in the decoding phase produces the most possible word based on the probability vocabulary distribution. However, it is possible the nal output sentence not to summarize the input document e ciently but a least similar word to have a better overall result. For example, the input document is: \Jane is excited about her upcoming journey to Africa next August since it was a dream comes true." and two possible summaries are \Jane is visiting Africa" and \Jane is going to visit Africa next August". The second possible summary is more informative about the input document. Nevertheless, when in the decoding phase the partial summary is \Jane is", the next most possible word comes to be \visiting" over \going" and as a result the nal summary will be the rst which is less informative. In order to solve this problem, the beam algorithm is a common solution that allows the model to generate *beam size* possible output summaries. If the *beam size* is equal to one then the nal summary will be based on the most similar word and in natural language processing this algorithm is named Gready. However, if *beam size* is greater than one on each step the *beam size* number partial output summaries will be those which maximize the following average log probability function.

$$\arg\max_{w} \prod_{w=1}^{N} \frac{1}{N} P(w^t | x; w^1; \ldots; w^{t-1})$$

where N is the total number of words in the summary and x is the input document.

3.3 Attention Mechanism

The attention mechanism is very successful through the various tasks of natural language processing, such as machine translation or text summarization. The reason behind its success in text summarization tasks is that the decoder does not take into consideration only the encoded representations, for example, the nal hidden and cell states of the source article as input, but also selectively focuses on parts of the article at each decoding step. To elaborate it, we are going to present the same example as the Abigail See's paper [52], which is the one from where we get our single document model. As it is easily observable in Figure 3.2 which presents the single document combined with the attention mechanism, the input sentence is \Germany emerges victorious in 2-0 win against Argentina on Saturday", while the decoder has already output as a summary the word \Germany" and it is about to calculate the next word which is going to be \beats". In order to achieve to generate the proper word \Argentina", attention mechanism is going to help in this direction. It will compute over all the tokens in the encoding phase an attention distribution which lets the decoder know where to attend to produce a target token. Given the encoding hidden states (after the concatenation between forwarding and backward LSTM cells) $h^e = [h_1^e; \ldots h_J^e]$, the formula for attention distribution for the step t and

Figure 3.2: Attention mechanism on Single Document Text Summarization Model [52]

the encoding token j will be the following:

$$a^e_{tj} = \frac{\exp s^e_{tj}}{\sum_{k=1}^{J} \exp s^e_{tk}}$$

The s_{tj} is an align score of a function in the step t for the j encoding token. The function is commonly given by this formula, where $v; W_{align}$ and b_{align} are learnable parameters:

$$s_{tj} = v^T \tanh\left(W_{align} \left(h^e_j \quad h^d_t \right) + b_{align}\right)$$

After computing the attention distribution it is time to compute the context vector. The context vector's job is to compress all the information from the attention distribution into a vector, which will be used in order to calculate the probability vocabulary distribution. The formula of context vector is the following :

$$z^e_t = \sum_{j=1}^{J} a^e_{tj} \quad h^e_j$$

Together with the current decoder hidden state h^d_t, we get the attention decoder hidden state:

$$\hat{h}^d_t = W_z \quad (z^e_t \quad h^d_t) + b_z$$

Finally, the probability vocabulary distribution, which will return us the word \beats" as the most possible and the next hidden state of the decoder will take their nal form:

$$P_{vocab;t} = softmax(W_{vocab} \quad \hat{h}^d_t + b_{vocab})$$

$$h^d_{t+1} = LSTM(h^d_t; E_{yt} \quad \hat{h}^d_t)$$

3.4 Pointer Generation Mechanism

The purpose of this mechanism is to allow the model to both copying words via pointing and generating words from the xed vocabulary. Implementing this mechanism is a proper solution to avoid out of vocabulary words. For example, if the

Figure 3.3: Pointer generation mechanism on Single Document Text Summarization Model [52]

input phrase is the previous one \Germany emerges victorious in 2-0 win against Argentina on Saturday" then the output phrase should be \Germany beats Argentina 2-0". Nevertheless, this speci c output phrase requires that the vocabulary holds the word \2-0". If the word \2-0" is not available the probability vocabulary distribution will give as output the most possible output based on the words it contains on it, i.g \3-2" and the nal summary would be wrong.

Figure 3.3 depicts the pointer generation mechanism. As it is easily observable, the only di erence between Figure 3.2 and 3.3 is the existence of a \switch", which determines whether to generate a token from the vocabulary or point to one in the source article at each decoding step. This \switch" is mathematically de ned as equation below shows, where the nal result is a scalar value,represents the sigmoid function and $W_z; W_h$ and b are learnable parameters.

$$p_{gen;t} = \quad (W_z \quad z^e_t + W_h \quad h^d_t + b)$$

When the \switch" is turn on the decoder produces a word from the vocabulary with the nal distribution $\hat{P}_{vocab;t}$. On the other hand, Otherwise, the decoder generates a pointer p_j based on the attention distribution where $p_j = \text{argmax}_{j=[1;:::;J]} a^e_{tj}$. This pointer p_j indicates the position of the token in the source article. As a result, the word embedding $E_{x(p_j)}$ of source article token will where the pointer indicates will be used as an input for the next decoding step.

3.5 Coverage Mechanism

Goal of this mechanism is to eliminate the problem of repetition in the nal summary. It is possible to repeat a word, which has been recently used in the summary, due to the fact that the attention mechanism utilizes the previous generated word in the summary text in order to calculate the attention distribution. A simple example is that in the previous partial generated summary phrase \Germany beats" the attention mechanism in order to generate the next word will take into consideration the word \beats". The word \beats" is strongly connected with both the words \Germany" and \Argentina" from the input source phrase \Germany emerges victorious in 2-0 win against Argentina on Saturday." As a result, it is possible the attention mechanism to choose \Germany" again as a correct output word, and the new partial generated summary will be \Germany beats Germany", which is totally wrong.

In order to solve this problem, the coverage mechanism proposes to penalize the words which have been used in the output summary when the attention distribution is calculated. More speci cally in the example when the attention distribution takes into consideration the output word \beats" then the word \Germany" which is also part of the partial summary will be penalized and will get a smaller value in the attention distribution so the model will prefer to choose the correct word \Argentina". Mathematically this is de ned as a coverage vector u_t^e, which represents the sum of attention distributions of the previous decoding steps.

$$u_t^e = \sum_{k=1}^{t-1} a_{tk}^e$$

This sum compresses the information about the attention information on each token in the source article during the previous decoding steps. About the initialization phase, $u_0^e = 0$ since none of the source document has been covered. The coverage vector will a ect the attention score as follows :

$$s_{tj} = v^T \tanh\left(W_{align} \begin{pmatrix} h_j^e & h_t^d & u_t^e \end{pmatrix} + b_{align}\right)$$

In conclusion, the attention at the current decoding time-step is aware of the attention during the previous decoding steps. Last but not least, a coverage loss will be calculated in each iteration in order to penalize the model if it is going to attend to the same locations when generating multi-sentence summaries. The coverage loss will have an upper bound of 1 and will be given as follows :

$$covloss_t = \sum_j \min(a_{tj}^e ; u_{tj}^e)$$

Chapter 4

Sentence Similarity Algorithm

In this chapter, the sentence-similarity algorithm will be presented. The sentence-similarity algorithm takes into consideration both the semantic and the syntactic nature of the observable sentences. One is responsible for the meaning of the words, and the other has to do with their order inside the sentence. The sum of them gives us the nal semantic similarity. A brief summarization of the proposed algorithm is depicted in Figure 4.1 below.

Figure 4.1: Proposed sentence similarity algorithm

The proposed algorithm uses a lexical database to compare the appropriate meaning of the word. Semantic similarity is calculated based on two semantic vectors. A semantic vector is formed for each sentence, which contains the weight assigned to each word for every other word from the second sentence in comparison. An order vector is formed for each sentence, which considers the syntactic similarity between the sentences. Finally, semantic similarity is calculated based on semantic vectors and order vectors. In sections 4.1 and 4.2 will be described in detail both the semantic and syntactic algorithm as long as the lexical database, Wordnet.

4.1 Semantic Similarity Algorithm

A quick de nition of the word \Semantics" is: Semantics is the linguistic and philosophical study of meaning in language, programming languages, formal logic, and semiotics. It is concerned with the relationship between signi ers like words, phrases, signs, and symbols|and what they stand for in reality, their denotation[35]. It has to do with the relationship between the words. As Figure 4.1 shows in order to calculate the sentence similarity, it is mandatory to calculate the similarity between the words. Word similarity is based on a lexical database called WordNet, which helps to disambiguate the meaning of each word on their sentences. The semantic algorithm is depicted on Algorithm 1.

Algorithm 1 Semantic similarity between sentences

```
1: procedure Semantic similarity
2:     $T_1$ - list of tokenized words    Sentence preprocessing
3:     $T_2$ - list of tokenized words    Sentence preprocessing
4:     $S_1$ - list of synsets    Word disambiguation($T_1$)
5:     $S_2$ - list of synsets    Word disambiguation($T_2$)
6:     vector length    max(length(S1),length(S2))
7:     $V_1; V_2$    vector length(null)
8:     $V_1; V_2$    vector length(path similarity(S1,S2))
9:     while  S1 do
10:        if path similarity value > Rubinstein threshold then
11:            $C_1$    $C_1 + 1$
12:    while  S2 do
13:        if path similarity value > Rubinstein threshold then
14:            $C_2$    $C_2 + 1$
15:    if sum($C_1; C_2$) = vector length=2 then
16:        Semantic similarity    $V_1$  $V_2$=vector length
17:    else
18:        Semantic similarity    cosine similarity($V_1; V_2$)
```

4.1.1 WordNet

WordNet is a lexical-semantic dictionary available for online and o ine use, developed and hosted at Princeton. The version used in this study is WordNet 3.0, which has 117,000 synonymous sets, Synsets. Synsets for a word represent the possible meanings of the word when used in a sentence. WordNet currently has the synset structure for nouns, verbs, adjectives, and adverbs. These lexicons are grouped separately and do not have interconnections; for instance, nouns and verbs are not interlinked.

The main relationship connecting the synsets is the super subordinate(ISA-HASA) relationship. The relation becomes more general as we move up the hierarchy. The root node of all the noun hierarchies is \Entity". Like nouns, verbs are arranged into hierarchies as well.

Figure 4.2: synsets for words: cat and dog

CatnDog	(dog.n.01)
(cat.n.01)	0.2
(guy.n.01)	0.125
(cat.n.03)	0.125
(kat.n.01)	0.07692
(caterpillar.n.02)	0.07692
(big_cat.n.01)	0.2
(computerized tomography.n.01)	0.05263

Table 4.1: Comparison between word \dog" and words in the synset of word \cat".

For example, the words \cat" and \dog" have a bunch of interpretations in their synsets as Figure 4.2 presents. In order to be clear the process of calculating the shortest path distance, an example will be given between the words \dog" and \cat" and in particular the interpretation of word \dog", \dog.n01." with all possible interpretations in the synset of word \cat". The Table 4.1 presents that the interpretation of word \dog", \dog.n01". has the shortest path distance with the interpretations \cat.n.01" and \big cat.n01." in the synonymous set of word \cat". The results are evident because, as humans, we have a stronger bond between the word \dog" and \cat" as a pet rather than with \cat" as computerized tomography. A stronger bond for Wordnet means the shortest path distance. At this point, it should be referred that Wordnet has a tree structure for synsets. On the root, there is a word named \subsumer", which is the common ancestor of its leaves. Wordnet has di erent tree-structure for words, which belong to a di erent part of the speech.

Hence, it is not possible to get a numerical value that represents the link between di erent parts of speeches like nouns and verbs. In order to reduce the time and space complexity of the algorithm, we only consider nouns, verbs, and adjectives to calculate the sentence similarity.

4.1.2 Sentence Preprocessing

Before calculating the semantic similarity between words, it is essential to determine which words will be utilized for comparison. For that reason, a word tokenizer and parts of speech tagging technique which were implemented in natural language processing toolkit NLTK were used. The rst move is to lowercase the capital letters in order to have the same form for all the words in the two compared sentences. The next move is to remove all the special characters and stopwords from the sentences, which are irrelevant to the semantic nature of the sentence. After this step, we lemmatize the words. Lemmatization's purpose is to reduce in ectional forms and sometimes derivationally related forms of a word to a common base form. For example words like \am",\are" and \is" are going to be converted to their base \be", which is known as the lemma The last step is to gure out which \part of speech" (POS) do the sentence's words belong to. This step lters the input sentence and tags the words into their \part of speech" (POS) and labels them accordingly. Table 4.2 presents all the parts of speech on an example sentence. This step is essential cause Wordnet has di erent structures for a di erent part of speech, so the information about the part of speech in a word allows us to categorize the word in the correct tree structure.

Word	Part of Speech
A	DT- Determiner
voyage	NN-Noun
is	VBZ-Verb
a	DT-Determiner
long	JJ-Adjective
journey	NN-Noun
on	IN-Preposition
a	DT-Determiner
ship	NN-Noun
or	CC- Coordinating conjunction
in	IN-Preposition
a	DT-Determiner
spacecraft	NN-Noun

Table 4.2: Parts of speeches

4.1.3 Word Disambiguation

After tokenization and tagging for each word with their part of speech, it is time for word disambiguation. Word disambiguation is the process according to which the algorithm understands which synset of the word better describes the meaning of it in the sentence. In the proposed algorithm, a method called \Lesk" (implemented on NLTK python library) was applied in order to nd the best- tted synset, which represents as close as possible the meaning of each word in their sentences. However, it should be pointed out that this algorithm is not the only one that was checked in the algorithm. \Max similarity" method was also implemented, but the results were not that good as the classic \Lesk" method. \Max similarity" method is implemented on Pywsd python library and calculates the path similarity between a selected word's synset and all the possible synsets of a second selected word. For instance, in the example with the words \dog" and \cat", the Max-similarity method would check the path similarity of synset \dog.n01" with all the synsets of the word cat, and it will choose as the best- tted synset the one with the maximum path similarity.

The \Lesk" method is based on the assumption that words in a given \neighborhood" (section of text) will tend to share a common topic. A simpli ed version of the \Lesk", method is to compare the dictionary de nition of an ambiguous word with the terms contained in its neighborhood. Versions have been adapted to use WordNet. An implementation might look like this:

> For every sense of the word being disambiguated, one should count the number of words that are in both neighborhoods of that word and in the dictionary de nition of that sense.
>
> The sense that is to be chosen is the sense which has the biggest number of this count.

A notorious example in word disambiguation eld is for the context of the word \bank". Figure 4.3 below we present two di erent synsets of the word \bank." According to the \Lesk" method, the best interpretation is the one with the most intersected words in the sentence and the de nitions given from Wordnet. In this case, it is the \bank.n01".

Figure 4.3: \Lesk" method example

4.1.4 Semantic Vectors

At the end of the word disambiguation part, two sets of synsets will have been created each for every sentence. These sets are essential in order to calculate the semantic vectors. A brief de nition of semantic vectors is that they are arrays in which there are values that depict the similarity between the synsets of the two sentences. The values have a range between [0,1]. The higher the value, the higher the similarity between the synsets. The similarity is measured by the distance on the tree-structured lexical database, Wordnet. Wordnet has a di erent tree for verbs and nouns. The di erent synsets of words are hanging on the leaves of the tree-structure. In Figure 4.4 below a small fragment of the noun, the tree structure is represented. For example, the distance between all the synsets for the word \motocycle" has a greater distance with the synsets of the word \bicycle" rather than the synsets of the word \car".

Figure 4.4: Hierarchical structure from WordNet

The calculation of semantic vectors starts with their initialization. Every unit of the semantic vector is initialized to null to void the foundational e ect. Below an example is presented in order to make clear how the semantic vectors get their values. Here are two candidate sentences:

S1: \A jewel is a stone used to decorate valuable things that you wear, such as rings or necklaces."

S2: \A gem is a jewel or stone that is used in jewellery."

After the word disambiguation phase there will be created two sets with synsets. Here are the sets for the above sentences.

Set1: [(stone.n.13), (use.v.01), (jewel.n.02), (necklace.n.01), (decorate.v.03), (thing.n.12), (valuable.a.01), (ring.n.08), (embody.v.02), (wear.v.09)]

Set2: [(jewelry.n.01), (stone.n.13), (use.v.01), (jewel.n.02), (embody.v.02), (gem.n.01)]

The length for each set is 10 and 6 respectively. The semantic vectors will have length equal to the maximum length of the two sets. Then, for each synset of set 1, the path similarity with all synset of set 2 will be calculated, and the similarity with the maximum score among all the similarities will be placed on a semantic vector called V_1. The path similarity between the synsets is ranged between 0 and 1. At this point, an allusion should be made that every result is above a standard threshold, a counter called C_1 would be increased. The threshold is called the Rubinstein threshold; it has a value of 0.8065, and according to Rubinstein 1965 [22], is the benchmark synonymy value. As a result, a semantic vector V_1 and a C_1 counter like these:

$$V1 = [0{:}5; 1{:}0; 1{:}0; 1{:}0; 1{:}0; 0{:}11; 0{:}0; 0{:}0; 0{:}0; 0{:}0]$$
$$C1 = 4$$

The same process will be repeated for each synset of set 2 respectively and as a result the second semantic vector called V_2 and a counter C_2 like these:

$$V2 = [1{:}0; 1{:}0; 1{:}0; 0{:}5; 0{:}1; 0{:}2; 0{:}0; 0{:}5; 1{:}0; 0{:}2]$$
$$C2 = 4$$

The motivation behind the idea of having two semantic vectors is quite simple. Every time we get the maximum value of all the similarities among all the synsets of a sentence for a target synset of the other sentence. It is possible that a couple of synsets between two sentences 1 and 2, for example, \jewelry-stone" has the maximum similarity from the perspective view of sentence 1 but the couple \stone-jewelry" may not have from the perspective view of sentence 2 since a di erent couple of synsets have the maximum path similarity. It is vital to keep this information to a di erent vector, the vector V_2, as well as a di erent counter C_2. A speci c example is depicted in Tables 4.3 and 4.4. Table 4.3 and Table 4.4 present all the path similarities for all the synsets of sentence 1 with all the synsets of sentence 2. With bold numbers are the maximum path similarity for a synset that has been tested with all the synsets of the other sentence. It is obvious to see that when the V_1 is lled, the couple of synsets(\gem.n.01")-(\thing.n.12") has the maximum path similarity equals 0.11, but the V_2 is lled, the couple of synsets (\thing.n.12")-(\jewel.n.02") has a path similarity equals 0.2 which is greater than the (\ thing.n.12")-(\gem.n.01") with path similarity equals 0.11. This is exactly the reason why there is a cross-comparison of the synsets and justify the existence of a second semantic vector.

In the end of \semantics vector" phase, two semantics vectors $(V_1; V_2)$ and two counters $(C_1; C_2)$ will have been created. The sentence similarity will be calculated as the cosine similarity between the two semantics vectors with a small twist in the calculation. When we compare two sentences that have almost none similar word, the semantics vectors will be lled with values very close to zero. There is a chance that these small values be the same for both semantics vector, and if we calculate the cosine similarity, the result will be close to one, which means that the sentences are similar, which is incorrect. To avoid this, we count the number of important words, and if their sum is lower than half of the vector's length, then the semantic similarity value will be calculated based on the multiplication of vectors V_1 and V_2 divided with their common vector length. This condition will help the nal sentence similarity algorithm to recognize non-similar sentences successfully.

Table 4.3

Sentence 1 - Sentence 2	Path Similarity
(jewelry.n.01)-(stone.n.13)	0.0714
(jewelry.n.01)-(use.v.01)	0
(jewelry.n.01)-(jewel.n.02)	0.11
(jewelry.n.01)-(necklace.n.01)	0.5
(jewelry.n.01)-(decorate.v.03)	0
(jewelry.n.01)-(thing.n.12)	0.125
(jewelry.n.01)-(valuable.a.01)	0
(jewelry.n.01)-(ring.n.08)	0.5
(jewelry.n.01)-(embody.v.02)	0
(jewelry.n.01)-(wear.v.09)	0
(stone.n.13)-(stone.n.13)	1
(stone.n.13)-(use.v.01)	0
(stone.n.13)-(jewel.n.02)	0.09
(stone.n.13)-(necklace.n.01)	0.067
(stone.n.13)-(decorate.v.03)	0
(stone.n.13)-(thing.n.12)	0.11
(stone.n.13)-(valuable.a.01)	0
(stone.n.13)-(ring.n.08)	0.067
(stone.n.13)-(ring.n.08)	0.067
(stone.n.13)-(embody.v.02)	0
(stone.n.13)-(wear.v.09)	0
(use.v.01)-(stone.n.13)	0
(use.v.01)-(use.v.01)	1
(use.v.01)-(jewel.n.02)	0
(use.v.01)-(necklace.n.01)	0
(use.v.01)-(decorate.v.03)	0.1
(use.v.01)-(thing.n.12)	0
(use.v.01)-(valuable.a.01)	0.33
(use.v.01)-(ring.n.08)	0
(use.v.01)-(embody.v.02)	0.0833
(use.v.01)-(wear.v.09)	0.2
(jewel.n.02)-(use.v.01)	0
(jewel.n.02)-(jewel.n.02)	1
(jewel.n.02)-(necklace.n.01)	0.1
(jewel.n.02)-(decorate.v.03)	0
(jewel.n.02)-(thing.n.12)	0.2
(jewel.n.02)-(valuable.a.01)	0
(jewel.n.02)-(ring.n.08)	0.1
(jewel.n.02)-(embody.v.02)	0
(jewel.n.02)-(stone.n.13)	0.09
(jewel.n.02)-(wear.v.09)	0
(jewel.n.02)-(wear.v.09)	0
(embody.v.02)-(stone.n.13)	0.055
(embody.v.02)-(use.v.01)	0.0833
(embody.v.02)-(jewel.n.02)	0
(embody.v.02)-(necklace.n.01)	0
(embody.v.02)-(decorate.v.03)	0.052
(embody.v.02)-(thing.n.12)	0
(embody.v.02)-(valuable.a.01)	0.0833
(embody.v.02)-(ring.n.08)	0
(embody.v.02)-(embody.v.02)	1
(embody.v.02)-(wear.v.09)	0.0714
(gem.n.01)-(stone.n.13)	0.067
(gem.n.01)-(use.v.01)	0
(gem.n.01)-(jewel.n.02)	0.09
(gem.n.01)-(necklace.n.01)	0.067
(gem.n.01)-(decorate.v.03)	0
(gem.n.01)-(thing.n.12)	0.11
(gem.n.01)-(valuable.a.01)	0
(gem.n.01)-(ring.n.08)	0.067
(gem.n.01)-(embody.v.02)	0
(gem.n.01)-(embody.v.02)	0
(gem.n.01)-(wear.v.09)	0

Table 4.4

Sentence 2 - Sentence 1	Path Similarity
(stone.n.13)-(jewelry.n.01)	0.0714
(stone.n.13)-(gem.n.01)	0.067
(stone.n.13)-(jewel.n.02)	0.09
(stone.n.13)-(stone.n.13)	1
(stone.n.13)-(use.v.01)	0
(stone.n.13)-(embody.v.02)	0
(use.v.01)-(jewelry.n.01)	0.1
(use.v.01)-(stone.n.13)	0.11
(use.v.01)-(use.v.01)	1
(use.v.01)-(jewel.n.02)	0
(use.v.01)-(embody.v.02)	0
(use.v.01)-(gem.n.01)	0
(jewel.n.02)-(jewelry.n.01)	0.11
(jewel.n.02)-(stone.n.13)	0.09
(jewel.n.02)-(use.v.01)	0
(jewel.n.02)-(jewel.n.02)	1
(jewel.n.02)-(embody.v.02)	0
(jewel.n.02)-(gem.n.01)	0.09
(necklace.n.01)-(stone.n.13)	0.067
(necklace.n.01)-(jewelry.n.01)	0.5
(necklace.n.01)-(use.v.01)	0
(necklace.n.01)-(jewel.n.02)	0.1
(necklace.n.01)-(embody.v.02)	0
(necklace.n.01)-(gem.n.01)	0.067
(decorate.v.03)-(stone.n.13)	0.0625
(decorate.v.03)-(jewel.n.02)	0.067
(decorate.v.03)-(use.v.01)	0.11
(decorate.v.03)-(embody.v.02)	0
(decorate.v.03)-(jewelry.n.01)	0.058
(decorate.v.03)-(gem.n.01)	0
(thing.n.12)-(jewelry.n.01)	0.125
(thing.n.12)-(stone.n.13)	0.11
(thing.n.12)-(use.v.01)	0
(thing.n.12)-(jewel.n.02)	0.2
(thing.n.12)-(embody.v.02)	0
(thing.n.12)-(gem.n.01)	0.11
(valuable.a.01)-(jewelry.n.01)	0
(valuable.a.01)-(stone.n.13)	0
(valuable.a.01)-(use.v.01)	0
(valuable.a.01)-(jewel.n.02)	0
(valuable.a.01)-(embody.v.02)	0
(valuable.a.01)-(gem.n.01)	0
(ring.n.08)-(jewelry.n.01)	0.5
(ring.n.08)-(stone.n.13)	0.067
(ring.n.08)-(use.v.01)	0
(ring.n.08)-(jewel.n.02)	0.1
(ring.n.08)-(gem.n.01)	0.067
(ring.n.08)-(embody.v.02)	0
(embody.v.02)-(jewelry.n.01)	0
(embody.v.02)-(stone.n.13)	0
(embody.v.02)-(use.v.01)	0.083
(embody.v.02)-(jewel.n.02)	0
(embody.v.02)-(embody.v.02)	1
(embody.v.02)-(gem.n.01)	0
(wear.v.09)-(jewelry.n.01)	0.083
(wear.v.09)-(stone.n.13)	0.09
(wear.v.09)-(use.v.01)	0.2
(wear.v.09)-(jewel.n.02)	0.1
(wear.v.09)-(embody.v.02)	0.0714
(wear.v.09)-(gem.n.01)	0.077

4.2 Syntactic Similarity Algorithm

Along with the semantic nature of the sentences, the syntactic structure of the sentences must be taken under consideration too. The syntactic similarity is the aggregation of comparisons of word indices in two sentences. The semantic similarity approach based on words, and the lexical database does not take into account the grammar of the sentence. The syntactic algorithm is depicted on Algorithm 2.

Algorithm 2 Syntactic similarity between sentences

```
 1: procedure Syntactic similarity
 2:     $T_1$ - list of tokenized words    Sentence preprocessing
 3:     $T_2$ - list of tokenized words    Sentence preprocessing
 4:     word index dict    union list($T_1; T_2$)
 5:     $r_1$    r ll vector (word index dict; $T_1$)
 6:     $r_2$    r ll vector (word index dict; $T_2$)
 7:     syntactic similarity    1   $\frac{r_1 \quad r_2}{r_1 + r_2}$
 8: procedure r fill vector    (word index dict, token list)
 9:     for word in word index dict do
10:         if word in token list then
11:             r vector    word index dict(word)
12:         else
13:             similar value, similar word    most similar word(word, token list)
14:             if similar value > Rubinstein threshold then
15:                 r vector    word index dict(similar word)
16:             else
17:                 r vector    0
```

For instance, consider a pair of sentences T_1 and T_2, that contain the same words in the same order except for two words from T_1, which occur in the reverse order in T_2. For example:

T_1: A quick brown dog jumps over the lazy fox.

T_2: A quick brown fox jumps over the lazy dog.

It is clear for a human interpreter that T_1 and T_2 are only similar to some extent. The dissimilarity between T_1 and T_2 is the result of the di erent word order. Therefore, a computational method for syntactic similarity should take into account the impact of word order. For the example pair of sentences T_1 and T_2, the joint word set is:

$$T = f\, A; quick; brown; dog; jumps; over; the; lazy; fox g$$

First and foremost, the sentences should be processed. Capital letters are lowercased, special characters are removed, and the words are lemmatized in order to receive their base form. The second step is to assign a unique index number for each word in T_1 and T_2. The index number is simply the order number in which the word appears in the sentence. For example, the index number is 4 for dog and 6 for over in T_1. In computing the syntactic similarity, a word order vector, r, is formed for T_1 and T_2, respectively, based on the joint word set $\mathbf{T}$. A word order vector is the basic structural information carried by a sentence. Taking T_1 as an example, for each word w_i in T, the procedure to nd the same or the most similar word in T_1 is the following:

1. If the same word is present in T_1, ll the entry for this word in r_1 with the corresponding index number from T_1. Otherwise, try to nd the most similar word $\hat{w}_i$ in T_1. By most similar word means the word which its synset has the maximum path similarity among the synsets of all the other words on its sentence.

2. If the similarity between w_i and $\hat{w}_i$ is greater than the Rubinstein threshold , the entry of w_i in r_1 is lled with the index number of $\hat{w}_i$ in T_1.

3. If the above two searches fail, the entry of w_i in r_1 is 0.

Having applied the procedure, the word order vectors for T_1 and T_2 are r_1 and r_2, respectively. The r vectors for the example sentences T_1 and T_2 will be the following:

$$r_1 = [1; 2; 3; 4; 5; 6; 7; 8; 9]$$
$$r_2 = [1; 2; 3; 9; 5; 6; 7; 8; 4]$$

After the calculation of word order vectors, the nal syntactic similarity is given from the below equation, and it is obvious that it has a range of $[0; 1]$:

$$\text{syntactic similarity} = 1 \quad \frac{r_1 \quad r_2}{r_1 + r_2}$$

Having calculated both semantic and syntactic similarity, the sentence similarity is a linear combination of these two values. In particular, a balance factor chooses the percentage of contribution for semantic and syntactic similarity, respectively. Of course, the nal sentence similarity will be in the range $[0; 1]$. The following equation describes the nal sentence similarity. For our work, the balance factor will be close to $= 0{:}9$ since the semantic nature is more important than the syntactic one when two sentences are compared.

$$\text{sentence similarity} = \quad \text{semantic similarity} + (1 \quad) \text{ syntactic similarity}$$

Chapter 5

Multi-Document Model

In this chapter, we present the state-of-the-art multi-document model [49] and propose a new updated version based on the sentence similarity algorithm, which was described in Chapter 4.

5.1 Baseline Multi Document Model

The state-of-the-art multi-document model was recently published in 2018 [49], and it proposed an adaptation method on the single document text summarization model, which was described in Chapter 3. The adaptation method is called Maximal Marginal Relevance (MMR). This method tries to nd which sentences are the most representative from all the multiple documents in order to use them as an input in the single document model to construct the summary. Given the fact that the used single document model has incorporated the pointer generator mechanism (PG) and in combination with the MMR method, authors named this multi-document technique as PG-MMR. Figure 5.1 represents the model.

5.1.1 Importance Models

The MMR algorithm should identify the most representative sentences for a summary in a set of documents. The way to distinguish which sentence is important and which not is based on importance models. For the state-of-the-art multi-document text summarization model was used 3 di erent importance models.

Figure 5.1: PG-MMR multi-document text summarization model

The rst one is called \Cosine". In the \Cosine" model, all the multiple documents are treated as sparse TF-IDF vectors. Through the cosine similarity metric, the model calculates the importance of a sentence in its document. The second is called \SummRec." In the \SummRec,", they used a pre-trained regression model to identify the importance of the sentences. This regression model was trained on (sentence, score) pairs where the training data are obtained from a single document dataset called CNN/ DailyMail with 33.000 articles. The score measured the maximum common sub-sequence between the articles and their ground truth summaries. A better version of the \SummRec" was also used, and it was named \BestSummRec." This version outperforms the \SummRec" due to its better-trained model.

5.1.2 Maximal Marginal Relevance Algorithm

The Maximal Marginal Relevance (MMR) algorithm, which was introduced in [49], tries to identify representative sentences from a set of documents, and with the help of the pointer generator single document model fuses them into an abstract. The MMR algorithm is depicted on Algorithm 3.

Algorithm 3 The MMR algorithm for summarizing multi-document inputs [49].

```
    Input : SDS data; MDS source sentences Si
 1: Train the PG model on SDS data.
 2: . $I(S_i)$ and $R(S_i)$ are the importance and redundancy scores of the source sentence $S_i$
 3: $I(S_i)$   ImportanceModel$(S_i)$ for all source sentences
 4: $MMR(S_i)$   $I(S_i)$ for all source sentences
 5: Summary  fg
 6: t   index of summary words
 7: while t   $L_{max}$ do
 8:     Find $Sk_{k=1}^{K}$ with highest MMR scores
 9:     Compute $a_{t;i}^{new}$ based on $Sk_{k=1}^{K}$
10:     Run PG decoder for one step to get $w_t$
11:     Summary   Summary + $w_t$
12:     if $w_t$ is the period symbol then
13:         $R(S_i)$   $Sim(S_i; Summary); 8i$
14:         $MMR_{score}(S_i) =$   $I(S_i) + (1$   $)$  $R(S_i); 8i$
```

This approach of handling multiple documents does not require training on multi-document datasets since the single document model will take care of the summary. Based on that, the PG-MMR text summarization technique should only train the single document model. After training the model, the rst step is to calculate the importance of the sentences in the set of documents based on one of the previously described importance models and assigning the importance value on the MMR score. This score is a mathematical representation of the sentence importance. After calculating all the MMR scores and sorting them in increasing order, the algorithm, in combination with the pointer generator single document model, starts to construct the summary. All the documents are combined into one and used as an input in the encoder (until a maximum number of tokens). However, the algorithm points out the most important sentences based on the MMR score, and in order to allow the

PG model to e ectively utilize the K source sentences without retraining the neural model, they dynamically adjust the attention weights a_{ti} at test time. In particular, the attention weights a_{ti} will be the same as described in Chapter 3.3 but only for the selected sentences. All the other sentences will be assigned with zero. Figure 5.2 depicts graphically how the \muting" mechanism works and below the equation describes the new attention weights of the \muting" mechanism, where the S_k is the set with all the k selected sentences and a_{ti} represents the attention weight of token i in time t.

Figure 5.2: \Muting"attention mechanism on Maximal Marginal Relevance Algorithm [49]

$$a_{ti}^{new} = \begin{cases} a_{ti} & ,i\ 2\ f\ S_k g_{k=1}^{K} \\ 0 & ,\text{otherwise} \end{cases}$$

After muting the non-important sentences, the PG decoder runs one time to produce a single word to the partial written summary. If the word is a period \.", that means that the model has successfully produced a summary sentence. If the word is not a period \.", then the process will continue until the PG decoder produces a period \.". When it happens, then it is time to calculate the \redundancy" of the chosen sentences. The \redundancy" of the sentences depicts how related they are with the partial written summary. The more related they are, the more redundant they are for the rest summary, which is going to be created. In order to measure the \redundancy", the MMR algorithm calculates the ROUGE-L precision, which expresses the longest common subsequence between a source sentence and the partial written summary mathematically. Having calculated the \redundancy" of the selected sentences, the MMR scores will be recalculated based on the MMR scoring formula, as is described below. The is a balance factor between the importance and the redundancy of the sentence.

$$MMR_{score}(s_i) = \quad Sim(s_i; D) + (1 \quad) \quad max Sim_{s_j\ 2\ S}(s_i; s_j) \qquad (5.1)$$

S and D represent the summary and the documents set, respectively. is a balance factor, and s_i is a selective sentence from the document, while the s_j is a selective sentence from the summary.

After recalculating the MMR scores, it is possible the attention weights to be rearranged since some of the current selected sentences will not be selected anymore. After several iterations, the algorithm will end when the maximum decoding tokens L_{max} will be produced.

Figure 5.3: PG-MMR-SS multi-document text summarization model

5.2 Multi Document Model With Sentence Similarity Algorithm

One of the most signi cant problems of the PG-MMR text summarization model is the updating MMR score method. In every iteration step, a new MMR score will be calculated based on equation 1. The varying part of the MMR score is the \redundancy" part. In the multi-document text summarization model, the \redundancy" part is calculated based on the longest common subsequence between a document sentence and the partial written summary. Nevertheless, the longest common subsequence does not take into consideration the semantic and syntactic nature of the most representative sentences. As a result, the nal summaries will not produce a meaningful summary. In order to solve this problem, we proposed a new way to calculate the \redundancy" part. Instead of using the longest common subsequence, we use the sentence similarity algorithm, as described in Chapter 4. In particular, at every iteration step, the sentence similarity algorithm will get as input one of the most important document sentences and one by one all the sentences from the partial generated summary, and it will return as an output a scalar value. This scalar value represents the maximum sentence similarity between the document sentence and the most similar sentence from the partial generated summary. The new \redundancy" part will be updated with this scalar value. The updated version of the MMR algorithm called MMR-SS, where \SS" stands for sentence similarity, is depicted below on Algorithm 4, while the nal proposed PG-MMR-SS multi-document model is presented in Figure 5.3.

Algorithm 4 The MMR-SS algorithm for summarizing multi-document inputs.

```
Input : SDS data; MDS source sentences Si
 1: Train the PG model on SDS data.
 2: . $I(S_i)$ and $R(S_i)$ are the importance and redundancy scores of the source sen-
    tence $S_i$
 3: $I(S_i)$   ImportanceModel($S_i$) for all source sentences
 4: $MMR(S_i)$   $I(S_i)$ for all source sentences
 5: Summary  fg
 6: t   index of summary words
 7: while t   $L_{max}$ do
 8:     Find $Sk_{k=1}^{K}$ with highest MMR scores
 9:     Compute $a_{t;i}^{new}$ based on $Sk_{k=1}^{K}$
10:     Run PG decoder for one step to get $w_t$
11:     Summary   Summary + $w_t$
12:     if $w_t$ is the period symbol then
13:         $R(S_i)$   Sentence Similarity Algorithm($S_i$; Summary); 8i
14:         $MMR_{score}(S_i) =$   $I(S_i) + (1$   $)$  $R(S_i)$; 8i
```

Chapter 6

Experiments

In order to investigate the efficiency of the proposed algorithm, the first experiment was conducted, in which we compared the sentence-similarity algorithm with a recently published one [42] in 2018 based on Pearson correlation coefficient and how close are the results to human evaluation results. The goal of the second experiment is to investigate how effectively the proposed PG-MMR-SS text summarization technique produces summaries by comparing the baseline PG-MMR text summarization technique and the proposed one, PG-MMR-SS, on two multi-document datasets named DUC-2004 and TAC-2011.

6.1 Sentence Similarity Experiments

6.1.1 Sentence Pairs

The experiment was conducted based on a standard dataset, which has 65 sentence pairs originally measure by Rubenstein and Goodenough [22]. Scientists have used these data in many investigations over the years, and they have been established as a stable source of the semantic similarity measure. In the [42], authors provide us with the human evaluation of these 65 sentence pairs. During the experiment, only 60 out of 65 pairs were used since the pairs *17:coast-forest*, *24:lad-wizard*, *30:coast-hill*, *33:hill-woodland* and *39:brother-lad* are not considered. The reason for this elimination is that those words have more than one common or synonymous words in their definitions. Therefore, the overall semantic similarity is falsely higher than the correct mean human rating. Some of the sentence pairs with the given mean human results are included in the table below.

Sentence 1	Sentence 2	Mean Human Resutls	Mago2018 Results	Proposed Algorithm Results
Midday is 12 o'clock in the middle of the day.	Noon is 12 o'clock in the middle of the day.	0.9550	0.8726	0.99
A mound of something is a large rounded pile of it.	A stove is a piece of equipment which provides heat, either for cooking or for heating a room.	0.05	0.4968	0.03
A grin is a broad smile.	A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly.	0.485	0.8419	0.26
A furnace is a container or enclosed space in which a very hot re is made, for example to melt metal, burn rubbish or produce steam.	Fruit or a fruit is something which grows on a tree or bush and which contains seeds or a stone covered by a substance that you can eat.	0.0475	0.1388	0.05
When you make a journey, you travel from one place to another.	A voyage is a long journey on a ship or in a spacecraft.	0.36	0.76	0.17
Your brother is a boy or a man who has the same parents as you.	A monk is a member of a male religious community that is usually separated from the outside world.	0.0450	0.278	0.02
A boy is a child who will grow up to be a man.	A lad is a young man or boy.	0.58	0.898	0.7
A forest is a large area where trees grow close together.	Woodland is land with a lot of trees..	0.6275	0.477	0.54
A cemetery is a place where dead peoples bodies or their ashes are buried.	A mound of something is a large rounded pile of it.	0.0575	0.0862	0.041
Noon is 12 o'clock in the middle of the day.	String is thin rope made of twisted threads, used for tying things together or tying up parcels	0.6275	0.477	0.54
An autograph is the signature of someone famous which is specially written for a fan to keep.	Your signature is your name, written in your own characteristic way, often at the end of a document to indicate that you wrote the document or that you agree with what it says.	0.405	0.3146	0.39
A magician is a person who entertains people by doing magic tricks.	In legends and fairy stories, a wizard is a man who has magic powers.	0.355	0.55	0.247
A cemetery is a place where dead peoples bodies or their ashes are buried.	A graveyard is an area of land, sometimes near a church, where dead people are buried.	0.7725	1	0.8
An implement is a tool or other pieces of equipment.	A tool is any instrument or simple piece of equipment that you hold in your hands and use to do a particular kind of work.	0.59	0.89	0.55

Table 6.1: Comparison between proposed methodology results with paper and mean human results

(a) (b)

(c) (d)

Figure 6.1: Pearson correlation coe cients and statistical measurements between algorithms.

6.1.2 Experimental Results

The evaluation between the proposed algorithm and the paper's algorithm will be measured with the Pearson correlation coe cient and some other statistics results such as the example's number, which are close enough to the mean human sentence similarity results. Table 6.1 shows some of these examples between the selected sentences with the mean human sentence similarity, the paper's similarity named Mago2018, and the proposed algorithm's similarity. In Appendix Chapter, all the sentence-pairs are presented in tables A.13 and A.14.

Figure 6.1 presents not only the nal results of Pearson correlation coe cients but also some statistical measurements about how close is the proposed algorithm in comparison with the paper's algorithm to mean human results. The sub gures (a) and (b) present the Pearson correlation coe cients for both the algorithms. Mago's paper results have a slightly bigger strength of a linear association between mean human similarity results (0.84) than the proposed ones (0.78). In the other two sub gures (c) and (d), the proposed algorithm's results are presented with blue color while with orange color are presented the paper's algorithm results. In the (c) sub gure, the x-axis is divided into three mean human results value ranges: small-correlation (0,0.4], medium-correlation (0.4, 0.7] and high-correlation (0.7, 1], and we count the number of examples (y-axis) which are closer to the mean human results between the two algorithms. Last but not least, the (d) sub gure presents how the divergence range of examples between the mean human results values with the proposed algorithm and paper algorithm results, respectively. In the x-axis, we set the divergence in 6 di erent ranges. The ranges are [0, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5] and (0.5, 1]. In the y-axis are presented the number of examples that their divergence between the algorithm results and the mean human results are inside the range in the x-axis.

6.2 Text Summarization Experiments

6.2.1 Datasets

For comparing the proposed PG-MMR-SS text summarization technique with the baseline PG-MMR, we used two of the most famous multi-document datasets in natural language processing called DUC-2004 and TAC-2011, respectively. The organization which is responsible for all these datasets called NIST [43]. This organization tries to nd almost each year documents for helping researchers in the natural language processing eld. There are a few multi-document datasets given from NIST organization such as DUC-2003, TAC-2008, TAC-2010, or TAC-2011 in order to research for the NLP eld, but DUC-2004 and TAC-2011 were utilized since we had easy access to some of their topics.

The DUC-2004 dataset stands for \Document Understanding Conferences (DUC)". It contains 50 TDT English news clusters, 24 TDT Arabic news clusters, and 50 TREC English news clusters. TDT and TREC are types of programs in order to collect the required clusters for the dataset. The objective of the Topic Detection and Tracking (TDT) program is to develop technologies that search, organize and structure multilingual, news-oriented textual materials from a variety of broadcast news media. Each TDT cluster contains 10 news wire/paper stories. Of course, it is vital for calculating the ROUGE results of the nal summary to have some reference summaries, which also included on the DUC-2004 dataset. In our case, 40 TDT cluster with 4 human-written reference summaries each were utilized.

The Text Analysis Conference of 2011 (TAC-2011) has divided their data based on their tasks. The TAC-2011 dataset is divided into 3 tasks: the Guided Summarization, Automatically Evaluating Summaries Of Peers (AESOP) and Multiling Pilot task. In our case, we used the data from the Multiling Pilot task. Task aimed to quantify and measure the performance of multilingual, multi-document summarization systems. For that reason, they created a multilingual dataset, which is derived from publicly available WikiNews texts [50]. The dataset contains multi-document and multilingual source texts based on this news from 7 di erent languages: Arabic, Czech, English, French, Greek, Hebrew, and Hindi. Native speakers of each language have translated texts in other languages. For each language. The dataset contains 10 topics of 10 articles each, and for each topic, there are 3 human-written references. For this thesis, we used the 10 topics with their reference summaries in the English language.

6.2.2 Experimental Set Up

For the experimental setup, model parameters have to be de ned. In particular, for this experiment, a pretrained single document point generator (PG) model was used. This model is trained for single-document summarization using the CNN/Daily Mail dataset containing single news articles paired with summaries (human-written article highlights). The training set contains 287,226 articles. An article contains 781 tokens; on average, a summary contains 56 tokens. About the hyperparameters of the single document model, the learning rate was 0.15, and the beamsize was 4 and an initial accumulator value of 0.1, while the vocabulary size was 50.000 words. The coverage mechanism was also enabled. During training, the encoding steps were maximum 400, while at test time, they were maximum 10000. The dimension size of word embeddings was 128, and the dimensions of hidden encoder states were 256. The decoding steps, which represent the words of the summary, were limited to a maximum of 100 tokens for training and maximum/minimum 120/100 tokens at test time. About the parameters of the Maximum Marginal Relevance algorithm, and especially for the muting mechanism, $= 7$ sentences were utilized per iteration. Also, all the 3 importance models which are described in the previous chapters were utilized in order to have an overall picture of the ROUGE results of the proposed PG-MMR-SS text summarization algorithm. Last but not least, as it is referred to in Chapter 4, the sentence-similarity algorithm has a balance factor between the semantic and syntactic similarity part. In order to be consistent with both experiments, the threshold was assigned to $= 0.9$.

Fitting this multi-document dataset on the text summarization requires a pre-processing step. All the documents and reference summaries should be included in a single document separated by one blank line like the one on the following table (suppose we have 2 articles and 2 reference summaries for the sake of this example). Then all the articles were read in order to remove any special characters, tokenize every word of them and turn each word into its word embeddings. After that, the process of summarization starts as it was described in the previous chapter with the help of the proposed sentence algorithm.

sentence 1 article 1
...
sentence N article 1
sentence 1 article 2
...
sentence M article 2
<SUMMARIES>
sentence 1 reference summary 1
...
sentence L reference summary 1
sentence 1 reference summary 2
...
sentence K reference summary 1

6.2.3 Experimental Results

In order to evaluate how e ectively the proposed PG-MMR-SS text summarization technique produces concise summaries, we used 40 English clusters from the DUC-04 dataset and 10 topics from the TAC-11 and evaluated them with 5 di erent types of ROUGE metrics: ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-S and ROUGE-Su4. The tables 6.2 and 6.3 depict in detail all the average F1 score percentage of ROUGE evaluation for Cosine similarity, SummRec, and BestSummRec importance model respectively. In the Appendix Chapter, all the F1 score, recall and precision scores for all the topics from DUC-2004 and TAC-2011 are presented in detail on Table A.1 until A.12.

Average F1 scores (%)		R1	R2	RL	RS	R-SU4
Cosine	PG-MMR	30.93	5.17	27.15	5.16	9.51
	PG-MMR-SS	31.89	5.7	28.26	5.6	10.05
SummRec	PG-MMR	32.7	6.02	28.47	5.79	10.24
	PG-MMR-SS	33.11	6.9	29.37	6.33	10.85
BestSummRec	PG-MMR	34.22	8.04	30.61	7.74	12.22
	PG-MMR-SS	35.06	8.52	31.26	8.01	12.6

Table 6.2: ROUGE results on the DUC-04 dataset.

Average F1 scores (%)		R1	R2	RL	RS	R-SU4
Cosine	PG-MMR	31.74	7.25	27.59	6.9	11.15
	PG-MMR-SS	32.62	8.26	29.13	7.5	11.78
SummRec	PG-MMR	31.54	7.52	27.92	6.8	11.05
	PG-MMR-SS	31.6	8.27	28.07	7.58	11.67
BestSummRec	PG-MMR	33.55	9.81	29.35	9.06	13.22
	PG-MMR-SS	33.66	9.96	30.14	9.24	13.38

Table 6.3: ROUGE results on the TAC-11 dataset.

To enhance the validity of the automatic evaluation metrics, we also present the summaries produced from the PG-MMR-SS and the PG-MMR methods on a random multi-document example. The automatic summaries are also compared to a human reference for a chosen cluster of DUC-04 dataset named D30036t, which is given below in the form of bullet points. The sentences which contain part of the content of the human reference summary are lled with bold text. The more sentences lled with bold text, the more meaningful the summary.

1. Jose Saramago is the rst Portuguese-language writer and one of few communists to win the Nobel Prize for Literature.

2. It is the fourth straight year that a European won.

3. He is widely acclaimed for his imaginative allegories.

4. Three American researchers, Robert Furchgott, Louis Ignarro, and Ferid Murad, shared the 1998 Nobel Prize for Medicine for discovering how nitric oxide acts as a signal molecule in the cardiovascular system.

5. Their research led to new treatments for heart and lung diseases, shock, and impotence.

6. The deliberations surrounding the awards are secret; however, Stig Ramel, a former director of the Nobel Foundation, revealed the committee wanted Jimmy Carter to share the 1978 peace prize.

D30036t cluster's summary from PG-MMR technique

\Portuguese novelist Jose Saramago says he was delighted to win the award . The Swedish academy announces the latest Nobel literature laureate. The Swedish Academy says it gave the award to Saramago for work that with parables sustained by imagination, compassion and irony continually enables us to apprehend an illusory reality. There have been other Portuguese authors, like Fernando Pessoa, whose work would justify 1,000 Nobels speaking at a packed press conference in Madrid. A day after winning the Nobel prize for literature, Portuguese novelist Jose Saramago, whose capricious vision includes a section of Europe breaking o . Furchgott is a pharmacologist at the state university of New York in Brooklyn."

D30036t cluster's summary from PG-MMR-SS technique

\Portuguese novelist Jose Saramago says he was delighted to win the award. He is also the fourth successive European to win the prize. A day after winning the Nobel prize for literature, Portuguese novelist Jose Saramago insisted that while parables sustained by imagination, compassion and irony continually enables us to apprehend an illusory reality.Three American researchers on Monday won the Nobel medicine prize for discovering how nitric oxide acts as a signal molecule in the cardiovascular system . The prize for physiology or medicine went to Dr. Robert Furchgott".

6.3 Discussion

In the rst experiment, the results are quite promising about the e ciency of the proposed sentence similarity algorithm. First, the Pearson correlation coe cient is over 0.7 (more speci cally 0.78), which shows a strong association between the mean human results and ours. Observing the sub gures (a) and (b), It is easily noticeable that both algorithms are close to the human mean similarity results not only for similar but also for non-similar sentence pairs. Mago's Pearson correlation coe cient is slightly better than ours, so that means that the paper's algorithm results can form a line lightly better than our results. Nevertheless, this indicator does not answer the question if the paper's algorithm is closer than the proposed one to the human results for all the sentence pairs. In order to investigate it, some statistical measurements were plotted. The sub gure (c) presents that the proposed algorithm is closer to mean human results in more sentence-pairs examples than Mago's algorithm does, regardless of the mean human similarity value range. More speci cally, the proposed algorithm is better on 60 % of the total example sentence pairs. Some sentence-pair examples are presented in table 4.1, where the results of our methodology are considerably better than Mago's results. The sub gure (d) shows the divergence range between the results of both algorithms. The 63,4 %

of the proposed algorithm results for the given 60 sentence-pairs examples have a deviation lower than 0.1, while only the 48.33 % of Mago's results belong in the same divergence range. The rest sentence-pair examples have a deviation of up to 0.5 for both algorithms. This is a strong indicator that the proposed algorithm can e ciently and e ectively recognize the semantic and syntactic nature between two sentences.

System	ROUGE 1	ROUGE 2	ROUGE SU4
ext-SumBasic (2007)	29.48	4.25	8.64
ext-KLSumm (2009)	31.04	6.03	10.23
ext-LexRank (2004)	34.44	7.11	11.19
abs-Extract-Rewrite (2018)	28.9	5.33	8.76
abs-Opinosis (2010)	27.07	6.03	8.63
abs-PG-Original (2017)	31.43	6.03	10.01
PG-MMR-SS/ Cosine	31.89	5.7	10.05
PG-MMR-SS/ SummRec	33.11	6.9	10.85
PG-MMR-SS/ BestSummRec	35.06	8.52	12.06

Table 6.4: Comparison between PG-MMR-SS results with other multi-document text summarization techniques [49]

In the second experiment, the results are also auspicious about how e ectively can the PG-MMR-SS text summarization technique produces concise summaries in comparison with the baseline PG-MMR technique. Table 6.2 and 6.3 present the ROUGE evaluation metrics on datasets DUC-04 and TAC-11, respectively. For both datasets, the PG-MMR-SS technique is better than the baseline one in every ROUGE evaluation metric was utilized up to 1.5%. Table 6.4 presents the nal average F1 score results of other text summarization techniques for the whole DUC-2004 dataset. In comparison with our results, Despite the fact that we use 40 clusters instead of 50, the proposed technique PG-MMR-SS surpasses by a large margin some unsupervised extractive baselines, including SumBasic, KLSumm, and LexRank and some abstractive like Opinosis and Extract- Rewrite.o enhance the validity of the automatic evaluation metrics, we also present the summaries produced from the PG-MMR-SS and the PG-MMR methods on a random multi-document example. We present the reference summary of a news wire topic from the DUC-04 (D30036t) in order to evaluate if the proposed text summarization technique PG-MMR-SS produces a more meaningful summary than the baseline does. The topic describes the 1998 Nobel prices on literature and medicine, and it informs us about the winners of the competition. We noticed that the summary from the PG-MMR-SS text summarization technique covered adequately 3 out of 6 sentences from the reference summary, while the baseline summary covered the meaning from only one sentence. In particular, the proposed summary refers the winners of the Nobel price in literature and medicine correctly and gives some details about the literature winner (being the fourth consecutive European winner) while the baseline summary only gives information about the literature winner since all the other sentences are meaningless related to the topic.

Chapter 7

Conclusion

In this thesis, we are proposing a sentence-similarity algorithm in order to improve a multi-document text summarization technique called PG-MMR. Our motivation is to create concise summaries, and in this direction, the sentence-similarity algorithm plays an important role. The proposed algorithm takes into consideration the semantic and syntactic nature of document sentences, so the nal summary is going to be more meaningful in comparison with the baseline summary.

In section 2, we set all the appropriate background work for text summarization and present related work in this eld both for single and multiple documents as input. In particular, we present some important Natural Language Processing techniques such as text wrangling and word embeddings, which allow us to handle textual data as input. The Long Short Term Memory networks (LSTMs) and in general, the Recurrent Neural Networks (RNNs) are described in detail since they are the basic component of our text summarization architecture. Summarization categories, as long as some useful summarization applications are also presented. Moreover, the evaluation metrics for sentence similarity algorithm and text summarization models are also a part of this chapter. Last, but not least, we refer to related work both for single and multiple text summarization approaches.

In sections 3 and 5, we are describing in detail how the baseline multi-document text summarization technique PG-MMR works. This technique consists of a single-document text summarization technique and an algorithm called Maximal Marginal Relevance (MMR), which is responsible for choosing the most representable sentences from all the documents. The single document text summarization technique contains some mechanism in order to make e cient summaries such as the attention, point-generator, and coverage mechanisms, as they are referred to in sections from 3.3 to 3.5. The MMR algorithm chooses the most representable sentences based on an MMR score, which is the subtraction between the \importance" and the \redundancy" of the sentence with respect to the partial-written summary. The \importance" part is calculated based on machine-learning importance models, while the \redundancy" part is measured based on the longest common subsequence between the sentence and the partial-written summary. In our thesis, we are suggesting a new way to calculate the redundancy part based on a sentence-similarity algorithm, which will take into account the semantic and syntactic part of the tested sentence. Based on the sentence-similarity algorithm, we improve the PG-MMR multi-document text-summarization technique into a new one called PG-MMR-SS.

In section 4, both the semantic and syntactic similarity part of the sentence-similarity algorithm are presented. This algorithm is classi ed in a major category of sentence-similarity algorithms, in which the algorithms are based on a lexical database. Our database is called Wordnet. About the semantic part, the goal is to give a mathematical representation of how related are two tested sentences based on the closest interpretation of their words in each sentence with the help of WordNet. About the syntactic part, the goal is to give also a mathematical representation of the relation between the two tested sentences based on the position of the words in their sentences. The addition of these two parts balanced with a factor gives the nal sentence-similarity.

In section 6, we present the results of two experiments that were conducted in order to investigate the e ciency of the similarity-algorithm and the e ectiveness of the new proposed text summarization technique PG-MMR-SS in comparison with the baseline PG-MMR. n the rst experiment, the proposed sentence-similarity algorithm was compared with a recently published sentence-similarity on 60 sentence-pairs examples algorithm based on the Pearson correlation coe cient and statistical measurements. The results were promising about the algorithm's e ciency since the algorithm has a Pearson correlation coe cient over 0.7 with the mean human results of the sentence-pairs. Also, the statistical measurements showed that our methodology is more accurate since the algorithm's results are closer to the mean human results than the recently published algorithm's ones. In the second experiment, we evaluate the proposed PG-MMR-SS technique based on multi-document topics from two datasets named DUC-2004 and TAC-2011 with several automatic ROUGE metrics, and we notice that the PG-MMR-SS is better than the baseline technique in every ROUGE metric.

Future work includes the addition of a transformer layer in the single document model in order not only to produce multi-lingual summaries but also to allow the model to recognize words from di erent languages. The data for this extension in the model will be based on TAC datasets. Also, our goal is to write a paper based on this thesis and submit it to the upcoming IEEE-ICASSP conference.

Appendix A

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
D3011t	28.73	29.61	27.89	5.2	5.3	5.05	26.56	27.37	25.79	4.65	4.8	4.51	8.74	9.02	8.48
D3017t	26.26	27.35	25.26	2.496	2.56	2.39	24.08	25.07	23.16	3.06	3.19	2.94	7	7.3	6.72
D3020t	22.53	22.56	22.5	1.12	1.13	1.12	20.31	20.33	20.28	2.42	2.42	2.42	5.83	5.84	5.82
D3022t	26.09	26.67	25.53	2.2	2.25	2.15	24.19	24.72	23.67	3.26	3.33	3.19	7.09	7.25	6.93
D3024t	20.28	20.11	20.46	2.28	2.26	2.3	18.59	18.44	18.75	1.81	1.79	1.82	4.98	4.94	5.03
D3026t	26.83	27.35	26.33	2.74	2.79	2.69	23.03	23.48	22.61	3.81	3.89	3.74	7.67	7.83	7.53
D3027t	22.31	23.1	21.58	2.2	2.28	2.13	19.32	20	18.58	2.59	2.68	2.5	5.93	6.15	5.73
D3028t	31.42	31.94	30.91	3.31	3.37	3.26	26.78	27.22	26.34	4.91	5	4.83	9.43	9.59	9.27
D3029t	32.76	34.45	31.31	8.28	8.68	7.91	28.27	29.68	27.02	6.17	6.48	5.88	10.69	11.23	10.21
D3031t	29.15	29.27	29.03	4.09	4.11	4.08	24.29	24.39	24.19	4.41	4.43	4.39	8.43	8.47	8.39
D3033t	20.32	21.68	19.13	2.19	2.34	2.06	18.16	19.36	17.09	2.13	2.27	2	5.16	5.52	4.85
D3034t	26.98	27.63	26.36	5.63	5.76	5.49	23.64	24.22	23.1	4.95	5.07	4.83	8.74	8.96	8.53
D3036t	29.99	31.05	28.99	3.89	4	3.76	26.14	27.07	25.27	3.98	4.13	3.85	8.36	8.67	8.08
D3037t	26.79	27.6	26.03	2.41	2.49	2.3	24.4	25.14	23.71	3.78	3.9	3.67	7.69	7.93	7.46
D3038t	31.45	32.3	30.65	6.69	6.88	6.52	27.86	28.61	27.15	7.08	7.27	6.89	11.27	11.59	10.98
D3040t	34.88	36.16	33.68	9.92	10.29	9.57	31.88	33.05	30.79	9.35	9.71	9.02	13.7	14.22	13.22
D3042t	38.98	39.57	38.42	6.21	6.3	6.12	32.31	32.79	31.84	8.05	8.18	7.94	13.33	13.53	13.13
D3045t	27.16	26.83	27.5	1.11	1.1	1.12	23.87	23.58	24.17	2.61	2.58	2.64	6.69	6.6	6.78
D3046t	35.28	35.23	35.33	4.94	4.93	4.95	28.49	28.46	28.53	5.22	5.21	5.22	10.34	10.33	10.35
D3047t	36.58	38	35.25	8.39	8.72	8.08	31.65	32.88	30.5	6.37	6.63	6.16	11.38	11.84	10.96
D3048t	30.07	30.03	30.11	2.99	2.98	2.99	25.24	25.2	25.27	3.61	3.6	3.6	8.11	8.1	8.12
D3049t	36.87	37.57	36.2	6.43	6.56	6.32	31.83	32.43	31.25	6.3	6.43	6.18	11.51	11.73	11.29
D3050t	26.43	26.26	26.61	2.16	2.15	2.17	24.03	23.87	24.19	2.1	2.08	2.11	6.23	6.19	6.27
D3051t	38.21	39.38	37.1	11.23	11.58	10.9	31.71	32.68	30.79	8.74	9.02	8.48	13.63	14.06	13.22
D3053t	28.65	29.83	27.55	3.22	3.35	3.09	25.46	26.52	24.49	3.56	3.71	3.42	7.78	8.11	7.74
D3055t	39.02	40.11	38	9.08	9.33	8.84	33.38	34.3	32.5	8.64	8.88	8.4	13.77	14.16	13.4
D3056t	35.73	36.44	35.05	3.64	3.71	3.57	31.86	32.49	31.25	5.96	6.08	5.84	10.99	11.21	10.77
D3059t	38.94	40.17	37.77	11.33	11.7	10.99	38.37	39.6	37.23	8.06	8.32	7.81	13.33	13.77	12.92
D31001t	31.52	32.22	30.85	6.87	7.02	6.72	29.08	29.72	28.46	5.73	5.86	5.6	10.03	10.26	9.81
D31008t	20.06	19.89	20.23	3.66	3.63	3.69	16.16	16.02	16.29	1.67	1.66	1.69	4.69	4.65	4.73
D31009t	33.77	34.31	33.25	7.67	7.8	7.55	31.41	31.91	30.93	6.54	6.65	6.44	11.04	11.22	10.87
D31013t	26.98	26.72	27.25	1.41	1.39	1.42	23.92	23.69	24.16	3.34	3.31	3.37	7.41	7.33	7.48
D31022t	34.45	34.36	34.55	4.25	4.24	4.26	30.25	30.17	30.34	5.511	5.49	5.52	10.35	10.32	10.38
D31026t	34.62	35.04	34.21	7.54	7.63	7.45	29.83	30.19	29.47	7.98	8.08	7.88	12.56	12.72	12.41
D31031t	32.65	32.96	32.34	6.66	6.72	6.59	27.43	27.7	27.17	5.9	5.96	5.84	10.5	10.61	10.4
D31032t	29.44	30.33	28.61	3.48	3.59	3.38	27.32	28.14	26.55	3.67	3.79	3.56	8.1	8.35	7.86
D31033t	35.94	35.12	36.8	7.21	7.05	7.39	32.1	31.37	32.86	6.07	5.93	6.22	11.12	10.86	11.39
D31038t	36.73	37.03	36.44	7.32	7.38	7.26	31.64	31.89	31.38	6.15	6.2	6.1	11.27	11.36	11.18
D31043t	37.42	38.72	36.2	11.43	11.83	11.05	32.3	33.43	31.25	9.85	10.2	9.52	14.5	15.05	14.02
D31050t	34.9	35.23	34.57	4.07	4.11	4.03	28.99	29.27	28.72	6.21	6.28	6.15	11.15	11.26	11.04
Average F1 score	30.93			5.17			27.15			5.16			9.51		

Table A.1: Baseline PG-MMR text summarization experiment results on DUC-2004 with Cosine Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR-SS	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
D3011t	28.57	29.61	27.6	4.63	4.8	4.47	25.88	26.82	25	4.46	4.62	4.3	8.6	8.92	8.3
D3017t	21.02	21.65	20.43	1.68	1.73	1.63	19.92	20.51	19.35	1.66	1.71	1.61	4.85	5	4.71
D3020t	22.1	22.01	22.19	1.13	1.13	1.14	19.86	19.78	19.94	2.26	2.25	2.27	5.62	5.6	5.65
D3022t	24.86	25.28	24.46	3.31	3.37	3.26	22.95	23.33	22.58	3.67	3.74	3.61	7.27	7.4	7.15
D3024t	26.17	26.54	25.81	2.51	2.54	2.47	23.69	24.02	23.37	3.02	2.98	3.06	7	7.1	6.9
D3026t	26.02	26.52	25.53	2.47	2.51	2.5	22.49	22.93	22.07	2.74	2.85	2.69	6.74	6.88	6.62
D3027t	25.65	26.33	25	2.21	2.27	2.15	24.28	24.93	23.67	2.71	2.78	2.64	6.51	6.69	6.34
D3028t	30.49	30.83	30.16	3.33	3.37	3.3	27.75	28.06	27.45	3.47	3.51	3.43	8.02	8.11	7.93
D3029t	29.85	31.3	28.53	6.94	7.28	6.63	24.83	26.04	23.74	5.02	5.27	4.79	9.2	9.66	8.78
D3031t	29.15	29.27	29.03	4.09	4.11	4.08	24.29	24.39	24.19	4.41	4.43	4.39	8.43	8.47	8.39
D3033t	21.53	22.83	20.36	3.58	3.8	3.38	19.35	20.52	18.3	3.21	3.42	3.03	6.27	6.66	5.92
D3034t	27.16	27.35	26.97	6.3	6.34	6.25	25.74	25.93	25.56	4.8	4.84	4.77	8.61	8.67	8.54
D3036t	30.6	31.24	29.89	4.22	4.32	4.12	27.26	27.92	26.63	4.09	4.19	3.99	8.5	8.72	8.3
D3037t	27.44	28.42	26.53	2.4	2.49	2.32	25.07	25.96	24.24	3.43	3.56	3.32	7.51	7.79	7.26
D3038t	31.42	31.73	31.11	5.67	5.73	5.69	27.49	27.76	27.22	5.75	5.81	5.69	10.12	10.22	10.02
D3040t	30.96	31.92	30.05	7.76	8	7.53	28.22	29.1	27.39	7.42	7.66	7.2	11.38	11.75	11.04
D3042t	40.37	41.19	39.58	6.17	6.3	6.05	35.06	35.77	34.38	7.74	7.9	7.58	13.21	13.49	12.95
D3045t	23.74	23.58	23.9	1.38	1.37	1.39	21.01	20.87	21.15	1.47	1.46	1.48	5.29	5.26	5.33
D3046t	34.19	34.15	34.24	5.76	5.75	5.77	27.41	27.37	27.45	5.89	5.88	5.9	10.76	10.74	10.77
D3047t	34.7	35.31	34.12	9.1	9.26	8.95	30.46	31	29.95	6.73	6.85	6.61	11.49	11.7	11.29
D3048t	30.71	30.83	30.59	3.24	3.25	3.23	25.9	26.03	25.8	3.81	3.82	3.79	8.34	8.37	8.3
D3049t	37.6	38.92	36.36	7.12	7.38	6.89	32.11	33.24	31.06	6.09	6.31	5.88	11.33	11.73	10.94
D3050t	24.16	23.87	24.46	1.63	1.61	1.65	20.94	20.69	21.2	1.77	1.75	1.8	5.62	5.55	5.69
D3051t	35.34	36.03	34.68	9.42	9.6	9.24	30.69	31.29	30.11	7.42	7.57	7.28	12.18	12.43	11.95
D3053t	29.82	31.21	28.53	4	4.19	3.83	26.12	27.35	25	4.25	4.46	4.2	8.55	8.97	8.2
D3055t	42.24	42.74	41.75	9.49	9.6	9.38	36.77	37.2	36.34	9.04	9.16	8.94	14.71	14.89	14.53
D3056t	36.21	36.72	35.71	3.66	3.71	3.61	31.48	31.92	31.04	6.11	6.2	6.02	11.15	11.31	10.99
D3059t	42.86	44.22	41.58	12.18	12.57	11.81	40.34	41.62	39.13	8.64	8.92	8.37	14.44	14.91	13.99
D31001t	33.7	34.44	32.98	7.14	7.3	6.99	28.26	28.8	27.66	6.01	6.15	5.88	10.73	10.97	10.49
D31008t	32.49	32.04	32.96	10.48	10.33	10.63	27.73	27.35	28.12	8.81	8.69	8.94	12.8	12.62	12.99
D31009t	39.42	39.63	39.2	10.96	11.02	10.9	36.24	36.44	36.05	8.741	8.79	8.7	13.88	13.96	13.81
D31013t	36.44	36.09	36.8	9.28	9.19	9.38	32.55	32.23	32.86	8.63	8.55	8.72	13.38	13.25	13.51
D31022t	36.66	37.99	35.42	8.18	8.48	7.9	31.27	32.4	30.24	8.19	8.5	7.9	13.09	13.58	12.63
D31026t	35.15	35.58	34.74	6.19	6.27	6.12	32.22	32.61	31.84	7.98	8.08	7.88	12.61	12.77	12.45
D31031t	32.83	32.96	32.69	6.14	6.16	6.11	28.41	28.53	28.3	7.07	7.11	7.04	11.51	11.56	11.46
D31032t	31.9	32.51	31.32	1.9	1.93	1.86	29.49	30.05	28.95	3.66	3.73	3.59	8.46	8.63	8.3
D31033t	37.18	36.73	37.64	6.58	6.5	6.67	33.65	33.24	34.07	8.02	7.92	8.13	12.95	12.79	13.11
D31038t	41.16	42.16	40.21	10.67	10.93	10.42	37.73	38.65	36.86	9.86	10.1	9.63	15.16	15.54	14.8
D31043t	36.07	37.33	34.9	8.98	9.3	8.68	33.38	34.54	32.29	7.57	7.84	7.31	12.42	12.87	12.01
D31050t	37.52	37.67	37.37	6	6.03	5.98	32.12	32.25	31.99	8.31	8.35	8.28	13.34	13.4	13.28
Average F1 score	31.89			5.7			28.26			5.6			10.05		

Table A.2: Proposed PG-MMR-SS text summarization experiment results on DUC-2004 with Cosine Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
D3011t	26.87	27.12	26.61	3.57	3.6	3.53	23.61	23.84	23.39	4.26	4.31	4.22	8.1	8.18	8.03
D3017t	30.83	32.12	29.64	2.71	2.83	2.6	28.69	29.89	27.58	3.49	3.64	3.35	8.05	8.4	7.73
D3020t	34.4	35.25	33.59	8.9	9.12	8.68	29.07	28.38	29.78	7.77	7.97	7.58	12.35	12.66	12.05
D3022t	28.73	28.61	28.85	5.53	5.51	5.56	26.27	26.16	26.37	3.96	3.94	3.98	8.13	8.09	8.16
D3024t	25.37	25.75	25	2.45	2.49	2.42	22.13	22.47	21.81	3.35	3.4	3.3	7.04	7.15	6.93
D3026t	33.55	34.42	32.73	6.94	7.12	6.77	30.91	31.71	30.15	6.82	7	6.65	11.37	11.67	11.09
D3027t	23.31	23.76	22.87	1.1	1.12	1.07	21.95	22.38	21.54	2.74	2.8	2.69	6.23	6.36	6.11
D3028t	35.05	36.24	33.93	9.05	9.37	8.76	32.41	33.52	31.38	7.02	7.27	6.79	11.75	12.16	11.36
D3029t	30.17	31.78	28.71	6.04	6.37	5.75	25.23	26.57	24.01	4.3	4.53	4.08	8.69	9.17	8.26
D3031t	32.67	32.987	32.37	7.25	7.32	7.18	29.48	29.76	29.21	6.97	7.04	6.9	11.34	11.45	11.24
D3033t	29.32	30.57	28.16	4.99	5.2	4.79	26.03	27.14	25	4.99	5.21	4.78	9.17	9.58	8.8
D3034t	29	29.86	28.19	5.53	5.7	5.38	25.17	25.91	24.47	3.9	4.02	3.79	8.13	8.37	7.89
D3036t	29.16	29.86	28.49	3.06	3.13	2.99	26.41	27.04	25.8	3.7	3.79	3.61	7.98	8.18	7.8
D3037t	32.98	33.24	32.71	6.78	6.83	6.72	31.37	31.62	31.12	5.65	5.7	5.6	10.3	10.39	10.22
D3038t	33.92	35.01	32.9	8.78	9.06	8.51	24.14	28.01	26.32	8.92	9.22	8.64	13.18	13.62	12.77
D3040t	30.46	31.34	29.64	7.23	7.44	7.03	28.87	29.7	28.09	6.73	6.93	6.54	10.68	10.99	10.38
D3042t	34.17	35.23	33.16	5.31	5.48	5.15	30.22	31.16	29.34	6.19	6.39	6	10.95	11.3	10.62
D3045t	34.11	33.87	34.34	5.52	5.48	5.56	30.83	30.62	31.04	6.43	6.39	6.48	11.1	11.02	11.18
D3046t	37.39	38.75	36.11	9.51	9.86	9.18	32.16	33.33	31.06	8.37	8.68	8.07	13.36	13.86	12.89
D3047t	33.82	34.23	33.42	9.69	9.81	9.57	29.29	29.65	28.95	9.3	9.41	9.19	13.48	13.64	13.31
D3048t	27.5	27.91	27.11	4.59	4.66	4.52	25.37	25.75	25	3.09	3.14	3.04	7.19	7.3	7.08
D3049t	35.43	36.49	34.44	9.02	9.29	8.76	30.45	31.35	29.59	8.78	9.05	8.53	13.37	13.78	12.98
D3050t	26.83	26.79	26.86	4.03	4.02	4.03	23.9	23.87	23.94	3.84	3.84	3.85	7.79	7.78	7.8
D3051t	34,5	35,75	33,33	11,44	11,86	11,05	30,73	31,84	29,69	8,36	8,67	8,06	12,63	13,1	12,19
D3053t	33.33	34.15	32.55	4.58	4.7	4.47	29.87	30.6	29.17	5.68	5.82	5.54	10.16	10.41	9.91
D3055t	39.28	40.37	38.25	8.04	8.27	7.83	34.66	35.62	33.75	6.41	6.59	6.24	12.01	12.35	11.69
D3056t	27.95	28.81	27.13	1.94	2	1.88	25.48	26.27	24.73	3.06	3.16	2.97	7.24	7.48	7.03
D3059t	38	39.31	36.96	7.36	7.6	7.14	33.05	34.1	32.06	6.61	6.83	6.4	12.03	12.42	11.66
D31001t	34.95	36.04	33.93	9.56	9.86	9.28	31.27	32.25	30.36	8.47	8.74	8.21	12.93	13.35	12.54
D31008t	32.16	32.34	32	5.46	5.49	5.44	29.46	29.62	29.3	5.48	5.51	5.44	10.02	10.08	9.96
D31009t	31.91	31.91	31.91	8.33	8.33	8.33	29.25	29.25	29.25	6.37	6.37	6.37	10.63	10.63	10.63
D31013t	36.34	37.19	35.53	4.63	4.74	4.52	32.3	33.06	31.58	6.51	6.67	6.36	11.55	11.83	11.28
D31022t	39.62	41.06	38.28	9.26	9.6	8.95	33.15	34.36	32.03	8.13	8.44	7.85	13.41	13.92	12.95
D31026t	30.46	31	29.95	6.16	6.27	6.05	27.55	28.03	27.08	6.89	7.02	6.77	10.9	11.1	10.71
D31031t	32.67	34.07	31.38	5.1	5.32	4.9	25.76	26.87	24.75	5.71	5.96	5.47	10.25	10.7	9.83
D31032t	32.44	33.06	31.84	5.42	5.53	5.312	30.83	31.42	30.26	4.6	4.698	4.51	9.38	9.57	9.21
D31033t	26.47	26.01	26.94	1.93	1.9	1.94	21.83	21.45	22.22	2.77	2.72	2.82	6.7	6.58	6.82
D31038t	34.83	35.68	34.02	5.87	6.01	5.73	32.19	32.97	31.44	8.06	8.27	7.87	12.58	12.89	12.28
D31043t	37.08	38.16	36.05	9.03	9.3	8.78	33.83	34.82	32.9	7.05	7.26	6.85	12.12	12.49	11.78
D31050t	28.19	28.46	27.93	4.34	4.38	4.3	26.04	26.29	25.8	3.5	3.53	3.46	7.74	7.81	7.66
Average F1 score	32.07			6.02			28.47			5.79			10.24		

Table A.3: Baseline PG-MMR text summarization experiment results on DUC-2004 with SummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR-SS	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
D3011t	24.43	24.93	23.95	3.53	3.6	3.46	23.09	23.56	22.63	3.88	3.97	3.8	7.42	7.57	7.27
D3017t	27.35	28.49	26.29	4.06	4.24	3.91	24.4	25.42	23.45	3.66	3.82	3.51	7.68	8.01	7.38
D3020t	34.13	34.97	33.33	7.55	7.74	7.37	29.6	30.33	28.91	6.56	6.72	6.4	11.21	11.49	10.94
D3022t	25.3	25.61	25	3.54	3.58	3.5	22.88	23.16	22.61	3.62	3.66	3.57	7.34	7.44	7.25
D3024t	22,19	21,92	22,47	2,8	2,77	2,84	19,97	19,73	20,23	2,76	2,72	2,79	6,1	6,02	6,18
D3026t	33.55	34.42	32.73	6.94	7.12	6.77	30.91	31.71	30.15	6.82	7	6.65	11.37	11.67	11.09
D3027t	23.31	23.76	22.87	1.1	1.12	1.07	21.95	22.38	21.54	2.74	2.8	2.69	6.23	6.36	6.11
D3028t	29.34	29.7	28.99	6.26	6.34	6.18	26.11	26.43	25.8	5.79	5.86	5.71	9.84	9.96	9.72
D3029t	35.74	37.26	34.34	9.3	9.69	8.93	30.22	31.51	29.04	6.89	7.2	6.62	11.76	12.28	11.29
D3031t	31.18	31.64	30.73	4.27	4.34	4.21	26.95	27.35	26.56	5.18	5.26	5.11	9.56	9.71	9.42
D3033t	29.04	30.29	27.89	6.09	6.36	5.85	27.12	28.29	26.05	4.7	4.91	4.51	8.75	9.14	8.39
D3034t	32.01	32.96	31.12	5.26	5.41	5.11	27.09	27.89	26.33	4.07	4.2	3.96	8.83	9.1	8.58
D3036t	34.3	34.93	33.7	11.75	11.97	11.54	30.43	30.99	29.89	9.44	9.62	9.27	13.73	13.99	13.48
D3037t	33.33	32.7	33.99	7.52	7.38	7.67	31.13	30.54	31.74	6.32	6.2	6.45	10.88	10.67	11.1
D3038t	31.77	33.33	30.36	7.56	7.93	7.22	27.24	28.52	26.02	7.95	8.35	7.58	11.91	12.51	11.36
D3040t	32.3	32.7	31.91	8.43	8.54	8.33	28.8	29.15	28.46	7.9	8	7.8	12.01	12.16	11.86
D3042t	34.08	34.96	33.25	6.41	6.58	6.25	29.85	30.62	29.12	6.28	6.44	6.12	11.01	11.3	10.73
D3045t	35.55	35.5	35.6	5.21	5.2	5.22	33.11	33.06	33.15	6.62	6.61	6.63	11.41	11.39	11.43
D3046t	37.39	38.75	36.11	9.51	9.86	9.18	32.16	33.33	31.06	8.37	8.68	8.07	13.36	13.86	12.89
D3047t	40.48	40.97	40	10.23	10.35	10.11	34.09	34.5	33.68	10.51	10.64	10.38	15.49	15.68	15.3
D3048t	28.46	28.73	28.19	4.07	4.11	4.03	25.24	25.47	25	3.11	3.14	3.08	7.37	7.44	7.3
D3049t	32.89	33.51	32.29	5.9	6.01	5.79	29.18	29.73	28.65	7.12	7.26	6.99	11.56	11.78	11.34
D3050t	30.34	29.97	30.71	5.7	5.63	5.77	26.31	26	26.63	4.55	4.49	4.61	8.94	8.83	9.05
D3051t	30,79	31,56	30,05	10.74	11.02	10.48	26.7	27.37	26.06	8.17	8.38	7.97	12.07	12.38	11.77
D3053t	31.71	37.97	31.45	5.48	5.53	5.44	28.73	28.96	28.46	5.38	5.42	5.33	9.81	9.9	9.73
D3055t	41.03	41.95	40.15	9.91	10.13	9.69	37.16	38	36.36	8.2	8.39	8.02	13.8	14.12	13.49
D3056t	32.06	33.05	31.12	6.37	6.57	6.18	29.86	30.79	28.99	5.67	5.85	5.49	9.97	10.29	9.67
D3059t	37.05	38.44	35.75	11.27	11.7	10.87	32.87	34.1	31.72	7.67	7.96	7.39	12.73	13.2	12.27
D31001t	35.33	36.04	34.63	10.2	10.41	10	31.87	32.52	31.25	9	9.19	8.82	13.39	13.67	13.13
D31008t	32.16	32.34	32	5.46	5.49	5.44	29.46	29.62	29.3	5.48	5.51	5.44	10.02	10.08	9.96
D31009t	38.77	38.56	38.98	10.27	10.22	10.33	34.49	34.31	34.68	8.67	8.63	8.72	13.62	13.55	13.7
D31013t	32.55	32.23	32.86	4.78	4.74	4.83	30.88	30.58	31.18	4.83	4.79	4.88	9.41	9.32	9.51
D31022t	40.6	41.62	39.63	10.19	10.45	9.95	35.15	36.03	34.31	8.56	8.79	8.35	14.08	14.44	13.73
D31026t	38.08	38.54	37.63	10.23	10.35	10.11	34.09	34.5	33.68	9.3	9.41	9.19	14.16	14.34	13.99
D31031t	32.39	33.24	31.58	4.64	4.76	4.52	26.45	27.15	25.79	5.69	5.85	5.54	10.19	10.47	9.93
D31032t	29.82	30.87	28.83	4.27	4.42	4.12	27.44	28.42	26.53	3.65	3.79	3.53	8.14	8.44	7.87
D31033t	28.68	28.95	28.42	1.88	1.9	1.86	23.9	24.13	23.68	3.73	3.77	3.7	7.97	8.05	7.9
D31038t	36.68	37.57	35.83	8.27	8.47	8.07	33.77	34.59	32.99	7.36	7.54	7.18	12.31	12.62	12.01
D31043t	37.28	38.16	36.44	9.08	9.3	8.87	32.38	32.15	31.65	6.64	6.8	6.48	11.86	12.15	11.59
D31050t	37.58	37.94	37.23	5.97	6.03	5.91	32.48	32.79	32.18	6.6	6.67	6.54	11.88	12	11.77
Average F1 score	33.11			6.9			29.37			6.33			10.85		

Table A.4: Proposed PG-MMR-SS text summarization experiment results on DUC-2004 with SummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
D3011t	29.53	30.14	28.85	4.34	4.43	4.25	25.77	26.3	25.26	4.11	4.19	4.02	8.48	8.66	8.3
D3017t	25.63	25.42	25.85	1.99	1.98	2.01	22.25	22.07	22.44	2.56	2.54	2.59	6.49	6.5	6.51
D3020t	27.88	28.42	27.37	2.71	2.76	2.66	24.13	24.59	23.68	4.76	4.86	4.67	8.69	8.87	8.53
D3022t	32.65	32.7	32.61	7.43	7.44	7.42	29.12	29.15	29.08	6.98	6.99	6.97	11.4	11.41	11.38
D3024t	18.31	18.08	18.54	1.96	1.94	1.99	18.03	17.81	18.26	3.33	3.29	3.37	5.91	5.83	5.99
D3026t	34.09	34.42	33.78	5.97	6.03	5.91	30.34	30.62	30.05	6.44	6.5	6.37	11.01	11.12	10.9
D3027t	27.06	28.18	26.02	1.61	1.68	1.55	23.87	24.86	22.96	3.23	3.37	3.11	7.28	7.59	6.99
D3028t	39.57	40.05	39.1	13.06	13.22	12.9	34.19	34.61	34.78	9.96	10.08	9.84	15.06	15.25	14.87
D3029t	33.33	34.78	32	10	10.44	9.6	30.73	32.06	29.5	8.39	8.76	8.04	12.63	13.2	12.11
D3031t	28.04	28.19	27.89	7.75	7.8	7.71	24.6	24.73	24.47	6.39	6.43	6.36	10.03	10.08	9.97
D3033t	26.24	26.35	26.12	4.85	4.87	4.83	24.82	24.93	24.72	5.31	5.34	5.29	8.92	8.96	8.88
D3034t	26.85	27.37	26.34	6.37	6.5	6.25	24.38	24.86	23.92	5.84	5.95	5.72	9.45	9.64	9.27
D3036t	35.18	35.48	34.89	12.05	12.15	11.94	32.13	32.4	31.87	9.17	9.25	9.09	13.51	13.63	13.4
D3037t	35.44	35.39	35.48	8.14	8.13	8.15	31.41	31.37	31.45	6.88	6.87	6.89	11.7	11.68	11.72
D3038t	34.41	35.56	33.33	8.97	9.27	8.68	31.99	33.06	30.99	10.06	10.4	9.73	14.21	14.69	13.75
D3040t	34.21	33.88	34.55	6.47	6.41	6.53	31.99	31.68	32.3	8.34	8.26	8.43	12.76	12.89	12.63
D3042t	43.79	44.44	43.16	12.42	12.6	12.23	38.98	39.57	38.42	12.97	13.16	12.77	18.09	18.37	17.81
D3045t	34.68	35.44	33.95	8.97	9.17	8.78	32.8	33.52	32.11	9.33	9.54	9.13	13.61	13.91	13.31
D3046t	36.19	37.13	35.31	10.15	10.41	9.9	32.23	33.06	31.44	9.71	9.97	9.47	14.23	14.61	13.87
D3047t	41.99	42.05	41.94	11.16	11.17	11.14	37.15	37.2	37.1	11.4	11.42	11.39	16.49	16.51	16.47
D3048t	30.34	30.62	30.05	5.43	5.48	5.38	27.11	26.86	26.86	4.61	4.65	4.56	8.98	9.07	8.9
D3049t	36.74	37.84	35.71	7.16	7.38	6.96	31.76	32.7	30.87	7.1	7.32	6.89	12.06	12.43	11.71
D3050t	31.7	31.83	31.58	5.87	5.9	5.85	28.53	28.65	28.42	6.44	6.47	6.41	10.74	10.78	10.7
D3051t	33.43	33.52	33.33	10.99	11.02	10.95	29.53	29.61	29.44	9.11	9.13	9.08	13.25	13.29	13.22
D3053t	31.1	31.69	30.53	4.34	4.42	4.25	28.69	29.23	28.16	5.04	5.14	4.95	9.43	9.63	9.25
D3055t	46.68	47.23	46.13	14.23	14.4	14.06	41.72	42.22	41.24	13.03	13.19	12.87	18.69	18.91	18.46
D3056t	38.48	40.11	36.98	8.49	8.86	8.16	36.59	38.14	35.16	7.67	8.01	7.37	12.74	13.3	12.23
D3059t	36.13	37.28	35.05	11.33	11.7	10.99	33.33	34.39	32.34	8.41	8.68	8.15	13.19	13.62	12.78
D31001t	32.37	31.98	32.78	5.83	5.75	5.9	30.18	29.81	30.56	6.35	6.28	6.44	10.74	10.61	10.88
D31008t	35.37	35.04	35.71	8.53	8.45	8.61	31.29	31	31.59	8.16	8.08	8.24	12.75	12.63	12.88
D31009t	37.37	36.96	37.77	9.78	9.68	9.89	33.6	33.25	33.97	8.11	8.02	8.2	12.87	12.73	13.01
D31013t	37.55	38.02	37.1	9.9	10.03	9.78	33.47	33.88	33.06	9.34	9.46	9.22	14.2	14.38	14.02
D31022t	37.19	37.71	36.69	7.8	7.91	7.69	32.51	32.96	32.06	7.92	8.04	7.81	12.82	13	12.64
D31026t	32.75	32.61	32.88	7.39	7.36	7.42	29.5	29.38	29.62	8.06	8.02	8.09	12.26	12.21	12.31
D31031t	32.75	33.79	31.77	5.97	6.16	5.79	26.04	26.87	25.26	6.16	6.36	5.97	10.69	11.04	10.36
D31032t	36.51	36.61	36.41	8.82	8.84	8.79	33.79	33.8	33.7	8.06	8.08	8.03	12.86	12.9	12.83
D31033t	32.5	32.98	32.03	4.81	4.88	4.74	29.06	29.49	28.65	6.06	6.15	5.97	10.51	10.67	10.36
D31038t	40.48	40.81	40.16	12.74	12.84	12.63	36.46	36.76	36.17	9.92	10	9.54	15.18	15.31	15.05
D31043t	45.74	47.08	44.47	15.6	16.06	15.16	37.89	39	36.84	14.6	15.04	14.19	19.93	20.53	19.36
D31050t	38.63	39.84	37.5	10.09	10.41	9.79	32.59	33.6	31.63	10.09	10.42	9.79	14.96	15.44	14.51
Average F1 score	34.22			8.04			30.61			7.74			12.22		

Table A.5: Baseline PG-MMR text summarization experiment results on DUC-2004 with BestSummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR-SS	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
D3011t	34.26	35.34	33.25	6.44	6.65	6.25	29.48	30.41	28.61	5.1	5.27	4.95	9.98	10.3	9.67
D3017t	23.23	22.91	23.56	1.72	1.69	1.74	20.68	20.39	20.98	2.52	2.49	2.56	5.99	5.9	5.91
D3020t	26.93	27.6	26.3	1.35	1.38	1.32	23.47	24.04	22.92	4.02	4.12	3.93	7.87	7.68	8.07
D3022t	29.39	29.43	29.35	5.23	5.23	5.22	25.85	25.89	25.81	5.63	5.63	5.62	9.62	9.64	9.61
D3024t	30.23	30.69	29.79	7.91	8.03	7.8	29.69	30.14	29.25	7.14	7.25	7.03	11.12	11.29	10.95
D3026t	33.91	34.42	33.42	5.67	5.75	5.58	29.91	30.352	29.47	6.4	6.5	6.3	11.09	11.26	10.9
D3027t	28.57	29.28	27.89	3.27	3.35	3.19	25.61	26.24	25	4.96	5.09	4.84	8.93	9.16	8.71
D3028t	38.76	39.24	38.3	13.06	13.22	12.9	33.92	34.33	33.51	9.96	10.08	9.84	14.92	15.11	14.74
D3029t	28.34	28.8	27.89	7.57	7.69	7.45	25.67	26.09	25.26	5.36	5.45	5.27	9.13	9.28	8.98
D3031t	34.82	35.37	34.28	10.05	10.22	9.9	31.94	32.45	31.44	9.3	9.45	9.15	13.64	13.87	13.43
D3033t	32.72	32.86	32.58	8.27	8.31	8.24	29.9	30.03	29.78	7.3	7.33	7.27	11.68	11.73	11.63
D3034t	30.9	31.841	30	7.67	7.91	7.45	27.64	28.49	26.84	5.88	6.07	5.71	10.14	10.64	9.84
D3036t	31.48	31.56	31.39	8.17	8.19	8.15	29.53	29.61	29.44	7.32	7.34	7.3	11.43	11.47	11.4
D3037t	36.78	36.73	36.83	8.96	8.94	8.97	32.75	32.71	32.8	7.66	7.65	7.67	12.57	12.56	12.59
D3038t	37.37	38.61	36.2	9.51	9.83	9.21	33.87	35	32.81	11.89	12.3	11.5	16.24	16.79	15.71
D3040t	32.92	32.78	33.06	6.43	6.41	6.46	29.6	29.48	29.72	8.13	8.09	8.16	12.4	12.35	12.45
D3042t	45.93	46.61	45.26	12.42	12.6	12.23	41.66	42.28	41.05	14.84	15.07	14.62	20.16	20.46	19.86
D3045t	37.85	37.64	38.06	8.1	8.06	8.15	33.7	33.52	33.98	8.06	8	8.1	13.09	13.02	13.17
D3046t	36.19	37.136	35.31	10.15	10.41	9.9	32.23	33.06	31.44	9.71	9.97	9.47	14.23	14.61	13.87
D3047t	40.32	40.16	40.49	12.31	12.26	12.36	35.45	35.31	35.6	12.31	12.26	12.36	17.14	17.07	17.21
D3048t	31.34	31.98	30.73	6.17	6.3	6.05	28.42	29	27.87	5.49	5.6	5.38	9.89	10.09	9.69
D3049t	38.12	39.46	36.87	9.24	9.56	8.93	32.9	34.05	31.82	8.46	8.77	8.18	13.43	13.91	12.98
D3050t	38.95	39.52	38.4	9.25	9.38	9.11	32.42	32.89	31.96	9.12	9.26	8.99	14.07	14.29	13.87
D3051t	38.15	39.11	37.23	11.85	12.15	11.56	32.7	33.52	31.91	9.24	9.48	9.01	14.12	14.49	13.78
D3053t	30.67	31.42	29.95	5.66	5.8	5.53	29.6	30.33	28.91	5.23	5.37	5.11	9.51	9.76	9.29
D3055t	43.91	43.27	44.56	16.78	16.53	17.03	39.09	38.52	39.67	12.78	12.59	12.98	18.1	17.83	18.38
D3056t	37.87	39.27	36.58	8.26	8.57	7.92	36.51	37.85	35.26	8.56	8.89	8.26	13.47	13.98	13
D3059t	36.06	36.99	35.17	11.68	11.99	11.39	33.52	34.39	32.69	8.69	8.92	8.47	13.41	13.77	13.07
D31001t	35.09	35.23	34.95	9.82	9.86	9.78	33.2	33.33	33.06	8.2	8.24	8.17	12.74	12.79	12.68
D31008t	31.29	31	31.59	5.23	5.18	5.28	28.57	28.3	28.852	5.06	5.01	5.11	9.53	9.44	9.62
D31009t	36.17	36.17	36.17	10.75	10.75	10.75	32.98	32.98	32.98	8.35	8.35	8.35	13	13	13
D31013t	34.97	34.43	35.51	11.32	11.14	11.49	30.49	30.03	30.97	11.29	11.11	11.47	15.28	15.04	15.53
D31022t	39.84	41.06	38.68	7.95	8.19	7.71	31.98	32.96	31.05	8.46	8.73	8.21	13.86	14.3	13.45
D31026t	36.54	36.39	36.69	10.12	10.08	10.17	31.94	31.81	32.06	8.45	8.41	8.48	13.24	13.29	13.18
D31031t	31.51	32.69	30.41	4.32	4.48	4.17	25.63	26.59	24.74	4.97	5.16	4.79	9.48	9.85	9.14
D31032t	39.45	39.34	39.56	8.31	8.29	8.33	35.07	34.97	35.17	7.14	7.12	7.16	12.61	12.57	12.64
D31033t	33.64	34.32	32.99	4.25	4.34	4.17	29.96	30.56	29.38	6.35	6.48	6.22	10.95	11.18	10.73
D31038t	39.52	40.27	38.8	11.53	11.75	11.32	34.22	34.87	33.59	10.19	10.39	10	15.24	15.54	14.95
D31043t	38.22	39.55	36.98	10.34	10.7	10	32.84	33.98	31.77	9.51	9.86	9.19	14.41	14.93	13.93
D31050t	40.37	41.19	39.58	13.69	13.97	13.42	35.86	36.59	35.16	11.52	11.77	11.29	16.36	16.7	16.03
Average F1score	35.06			8.52			31.26			8.01			12.6		

Table A.6: Proposed PG-MMR-SS text summarization experiment results on DUC-2004 with BestSummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
0	31.62	32.14	31.12	9.84	10	9.68	29.19	29.67	28.72	8.44	8.58	8.3	12.34	12.55	12.13
1	32.42	32.78	32.06	5.83	5.9	5.77	28.02	28.33	27.72	5.91	5.98	5.84	10.42	10.54	10.31
2	35.23	35.52	34.95	9.31	9.39	9.24	31.44	31.69	31.18	9.19	9.27	9.11	13.68	13.79	13.56
3	31.13	32.29	30.05	3.9	4.05	3.76	24.24	25.14	23.4	5.13	5.32	4.95	9.6	9.97	9.26
4	35.21	34.5	35.95	8.34	8.17	8.52	29.71	29.11	30.34	7.45	7.3	7.62	12.19	11.93	12.45
5	27.76	27.79	27.72	3.58	3.58	3.57	23.94	23.98	23.91	3.82	3.83	3.82	7.89	7.9	7.88
6	24.97	24.65	25.28	2.84	2.8	2.87	20.48	20.22	20.74	2.73	2.69	2.77	6.51	6.42	6.59
7	37.47	39.27	35.83	13.62	14.29	13.02	35.31	37.01	33.76	13.15	13.8	12.55	17.3	18.15	16.52
8	34.77	34.87	34.68	11.17	11.2	11.4	30.19	30.27	30.11	8.19	8.21	8.17	12.77	12.8	12.73
9	26.72	26.12	27.35	6.69	6.53	6.84	25	24.44	25.59	6.55	6.4	6.71	9.98	9.75	10.22
Average F1 score	31.74			7.25			27.59			6.9			11.15		

Table A.7: Baseline PG-MMR text summarization experiment results on TAC-2011 with Cosine Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR-SS	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
0	27.72	28.02	27.42	6.32	6.39	6.25	24.46	24.72	24.19	5.62	5.68	5.56	9.38	9.48	9.27
1	30.91	31.94	29.95	5.16	5.34	5	27.69	28.61	26.82	5.78	5.98	5.59	10.01	10.35	9.69
2	35.58	36.07	35.11	6.54	6.63	6.45	32.88	33.33	32.45	8.41	8.53	8.3	13.04	13.23	12.86
3	32.59	33.43	31.79	7.04	7.22	6.87	28.69	29.43	27.99	6.92	7.1	6.74	11.34	11.64	11.05
4	38.22	38.27	38.17	11.16	11.17	11.14	33.11	33.15	33.06	8.85	8.86	8.83	13.81	13.83	13.79
5	28.97	29.09	28.85	4.46	4.48	4.44	24.83	24.93	24.72	4.05	4.07	4.03	8.29	8.33	8.25
6	37.87	39.27	36.58	12.95	13.43	12.5	35.15	36.44	33.95	12.62	13.1	12.17	16.98	17.62	16.38
7	34.42	34.32	34.51	13.7	13.66	13.74	30.62	30.54	30.71	8.29	8.27	8.32	12.7	12.66	12.73
8	27.3	26.68	27.94	6.98	6.82	7.14	24.71	24.16	25.29	7.2	7.03	7.38	10.47	10.23	10.73
9	27.72	28.02	27.42	6.32	6.39	6.25	24.46	24.72	24.19	5.62	5.68	5.56	9.38	9.48	9.27
Average F1 score	32.62			8.26			29.13			7.53			11.78		

Table A.8: Proposed PG-MMR-SS text summarization experiment results on TAC-2011 with Cosine Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
0	32.87	32.69	33.06	11.17	11.11	11.24	29.83	29.67	30	9.71	9.66	9.77	13.71	13.63	13.79
1	28.02	28.33	27.72	2.78	2.81	2.75	25.55	25.83	25.27	3.98	4.02	3.93	8.02	8.11	7.93
2	33.24	33.33	33.15	7.44	7.46	7.42	30.24	30.33	30.16	6.42	6.44	6.4	10.9	10.93	10.87
3	29.9	30.57	29.4	7.08	7.22	6.94	27.45	28	26.92	6.55	6.69	6.42	10..44	10.66	10.24
4	34.91	34.77	35.05	8.21	8.17	8.24	31.12	31	31.25	6.21	6.18	6.24	11.1	11.05	11.15
5	25.1	25.07	25.27	2.49	2.48	2.5	21.89	21.8	21.98	3.23	3.21	3.24	6.91	6.88	6.93
6	24.3	25.49	23.96	2.99	3.08	2.9	22.66	24.1	23.36	3.55	3.67	3.44	7.14	7.37	6.92
7	34.9	35.59	34.24	7	7.14	6.87	29.01	30.23	29.08	6.3	6.43	6.18	11.09	11.31	10.87
8	40.1	40.54	39.89	17.07	17.21	16.94	33.11	34.68	34.42	13.08	13.18	12.97	17.76	17.9	17.61
9	32.09	33.71	30.61	8.92	9.38	8.51	28.34	29.78	27.04	8.95	9.42	8.53	12.84	13.51	12,24
Average F1 score	31.54			7.52			27.92			6.8			11.05		

Table A.9: Baseline PG-MMR text summarization experiment results on TAC-2011 with SummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR-SS	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
0	34.24	34.62	33.87	11.54	11.67	11.41	30.16	30.49	29.84	9.78	9.89	9.67	13.85	14.01	13.7
1	29.01	29.17	28.85	3.63	3.65	3.61	24.59	24.72	24.45	4.34	4.37	4.32	8.49	8.54	8.44
2	23.81	23.22	24.42	3.4	3.31	3.49	22.13	21.04	21.57	2.26	2.2	2.32	5.92	5.77	6.08
3	30.7	31.14	30.28	7.12	7.22	7.02	26.76	27.14	26.39	6.88	6.98	6.78	11	11.15	10.83
4	38.53	39.62	37.5	11.13	11.44	10.82	32.77	33.69	31.89	9.26	9.53	9	14.25	14.66	13.86
5	29.77	29.77	29.57	6.57	6.61	6.52	26.25	26.43	26.07	8.17	8.23	8.11	11.8	11.88	11.72
6	25.37	26.04	24.74	2.18	2.24	2.13	22.94	23.55	22.37	3.29	3.38	3.21	6.9	7.09	6.72
7	31.78	32.77	30.85	8.03	8.29	7.8	28.49	29.38	27.66	7.65	7.9	7.42	11.76	12.14	11.4
8	40.7	40.81	40.59	20.16	20.22	20.11	38.27	38.38	38.17	15.17	15.75	15.67	19.93	19.99	19.88
9	32.09	33.71	30.61	8.92	9.38	8.51	28.34	29.78	27.04	8.95	9.42	8.53	12.84	13.51	12,24
Average F1 score	31.6			8.27			28.07			7.58			11.67		

Table A.10: Proposed PG-MMR-SS text summarization experiment results on TAC-2011 with SummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
0	29.57	30.22	28.95	5.98	6.11	5.85	26.88	27.47	26.32	5.28	5.4	5.16	9.32	9.53	9.12
1	34.41	35.56	33.33	13.04	13.48	12.63	30.38	31.39	29.43	14	14.48	13.55	17.44	18.03	16.88
2	23.42	23.22	23.61	3.06	3.04	3.09	21.21	21.04	21.39	2.22	2.2	2.24	5.82	5.77	5.87
3	38.53	38.86	38.2	16.05	16.18	15.91	33.43	33.71	33.15	13.67	13.79	13.55	17.82	17.98	17.66
4	33.29	33.15	33.42	7.39	7.36	7.42	25.98	25.88	26.09	5.71	5.68	5.73	10.4	10.36	10.45
5	28.8	29.15	28.46	4.63	4.68	4.57	26.38	26.7	26.06	6.06	6.14	5.99	9.93	10.06	9.81
6	32.76	34.35	31.31	5.34	5.6	5.1	27.48	28.81	26.26	6.39	6.7	6.09	10.83	11.37	10.34
7	37.19	38.14	36.29	8.36	8.57	8.15	32.51	33.33	31.72	8.83	9.06	8.61	13.72	14.08	13.38
8	39.24	38.92	39.56	18.4	18.31	18.67	36.24	35.95	36.54	14.59	14.47	14.72	18.85	18.69	19.01
9	34.32	35.67	33.07	12.02	12.5	11.58	30.54	31.74	29.43	10.11	10.52	9.73	14.19	14.77	13.66
Average F1 score	33.55			9.81			29.35			9.06			13.22		

Table A.11: Baseline PG-MMR text summarization experiment results on TAC-2011 with BestSummRec Importance model

		Rouge-1			Rouge-2			Rouge-L			Rouge-s4			Rouge-su4	
PG-MMR	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision	F- score	Recall	Precision
0	33.06	33.79	32.37	10.05	10.28	9.84	31.18	31.87	30.53	8.95	9.15	8.75	13.05	13.35	12.77
1	34.41	35.56	33.33	13.04	13.48	12.63	30.38	31.39	29.43	14	14.48	13.55	17.44	18.03	16.88
2	28.77	28.69	28.85	5.82	5.8	5.83	26.85	26.78	26.92	6.97	6.95	6.99	10.68	10.65	10.71
3	35.9	36	35.79	13.26	13.3	13.2	31.91	32	31.82	10.44	10.47	10.41	14.79	14.83	14.75
4	33.92	33.96	33.87	6.81	6.78	6.91	29.07	29.11	29.03	5.56	5.57	5.56	10.39	10.41	10.38
5	31.76	32.15	31.38	8.98	9.096	8.87	28.26	28.61	27.93	9.18	9.3	9.07	12.89	13.05	12.73
6	30.98	32.13	29.9	4.59	4.76	4.43	27.5	28.53	26.55	5.9	6.13	5.69	10.12	10.51	9.76
7	37.19	38.14	36.29	8.36	8.57	8.15	32.51	33.33	31.72	8.83	9.06	8.61	13.72	14.08	13.38
8	35.5	35.4	35.6	16.4	16.3	16.38	32.52	32.43	32.61	12.16	12.12	12.19	16.19	16.14	16.23
9	35.13	36.52	33.85	12.3	12.78	11.84	31.25	32.58	30.21	10.45	10.87	10.05	14.56	15.15	14.02
Average F1 score	33.66			9.96			30.14			9.24			13.38		

Table A.12: Proposed PG-MMR-SS text summarization experiment results on TAC-2011 with BestSummRec Importance model

Table A.13: Sentence-pairs with the results from the proposed, Mago2018 and mean human results

Sentence 1	Sentence 2	Mean Human Results	Mago2018 Results	Proposed Algorithm Results
Cord is strong, thick string	A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly	0.01	0.0225	0.009
A rooster is an adult male chicken.	A voyage is a long journey on a ship or in a spacecraft.	0.005	0.2593	0.037
Noon is 12 o'clock in the middle of the day.	String is thin rope made of twisted threads, used for tying things together or tying up parcels.	0.0125	0.034	0.033
Fruit or a fruit is something which grows on a tree or bush and which contains seeds or a stone covered by a substance that you can eat.	A furnace is a container or enclosed space in which a very hot re is made, for example to melt metal, burn rubbish or produce steam.	0.0475	0.1388	0.05
An autograph is the signature of someone famous which is specially written for a fan to keep.	The shores or shore of a sea, lake, or wide river is the land along the edge of it.	0.0050	0.07	0.17
An automobile is a car.	In legends and fairy stories, a wizard is a man who has magic powers.	0.02	0.0088	0.028
A mound of something is a large rounded pile of it	A stove is a piece of equipment which provides heat, either for cooking or for heating a room.	0.005	0.4968	0.0279
A grin is a broad smile.	An implement is a tool or other pieces of equipment.	0.005	0.0099	0.03
An asylum is a psychiatric hospital.	Fruit or a fruit is something which grows on a tree or bush and which contains seeds or a stone covered by a substance that you can eat.	0.005	0.1456	0.02
An asylum is a psychiatric hospital.	A monk is a member of a male religious community that is usually separated from the outside world.	0.0375	0.0175	0.04
A graveyard is an area of land, sometimes near a church, where dead people are buried.	If you describe a place or situation as a madhouse,you mean that it is full of confusion and noise.	0.0225	0.1339	0.038
When you make a journey, you travel from one place to another.	A voyage is a long journey on a ship or in a spacecraft.	0.36	0.7826	0.175
An autograph is the signature of someone famous which is specially written for a fan to keep.	Your signature is your name, written in your own characteristic way, often at the end of a document to indicate that you wrote the document or that you agree with what it says.	0.405	0.3146	0.389
The coast is an area of land that is next to the sea.	The shores or shore of a sea, lake, or wide river is the land along the edge of it.	0.5875	0.9773	0.152
A forest is a large area where trees grow close together.	Woodland is land with a lot of trees.	0.6275	0.477	0.54
An implement is a tool or other pieces of equipment.	A tool is any instrument or simple piece of equipment that you hold in your hands and use to do a particular kind of work.	0.59	0.8919	0.55
A cock is an adult male chicken.	A rooster is an adult male chicken.	0.8625	0.856	0.81
A boy is a child who will grow up to be a man.	A lad is a young man or boy.	0.58	0.898	0.7
A cushion is a fabric case lled with soft material, which you put on a seat to make it more comfortable.	A pillow is a rectangular cushion which you rest your head on when you are in bed.	0.5225	0.934	0.05
A cemetery is a place where dead peoples bodies or their ashes are buried.	A graveyard is an area of land, sometimes near a church, where dead people are buried.	0.7725	1	0.8
An automobile is a car.	A car is a motor vehicle with room for a small number of passengers.	0.5575	0.7	0.1
Midday is 12 oclock in the middle of the day	Noon is 12 oclock in the middle of the day	0.955	0.8726	0.99
A gem is a jewel or stone that is used in jewellery.	A jewel is a precious stone used to decorate valuable things that you wear, such as rings or necklaces.	0.6525	0.8536	0.58

Table A.14: Sentence-pairs with the results from the proposed, Mago2018 and mean human results

Sentence 1	Sentence 2	Mean Human Results	Mago2018 Results	Proposed Algorithm Results
A bird is a creature with feathers and wings, females lay eggs, and most birds can y.	A cock is an adult male chicken.	0.2425	0.1379	0.025
Food is what people and animals eat.	Fruit or a fruit is something which grows on a tree or bush and which contains seeds or a stone covered by a substance that you can eat.	0.045	0.278	0.023
Your brother is a boy or a man who has the same parents as you.	A monk is a member of a male religious community that is usually separated from the outside world.	0.215	0.186	0.0189
An asylum is a psychiatric hospital.	If you describe a place or situation as a madhouse, you mean that it is full of confusion and noise.	0.3475	0.1613	0.053
A furnace is a container or enclosed space in which a very hot re is made, for example, to melt metal, burn rubbish, or produce steam.	A stove is a piece of equipment which provides heat, either for cooking or for heating a room.	0.355	0.55	0.1928
A magician is a person who entertains people by doing magic tricks.	In legends and fairy stories, a wizard is a man who has magic powers.	0.2925	0.2986	0.05
A hill is an area of land that is higher than the land that surrounds it.	A mound of something is a large rounded pile of it.	0.47	0.253	0.192
Cord is strong, thick string.	String is thin rope made of twisted threads, used for tying things together or tying up parcels.	0.1375	0.3016	0.47
A grin is a broad smile.	A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly.	0.485	0.8419	0.258
In former times, serfs were a class of people who had to work on a particular persons land and could not leave without that persons permission..	A slave is someone who is the property of another person and has to work for that person.	0.4825	0.8896	0.4326
Glass is a hard transparent substance that is used to make things such as windows and bottles.	A magician is a person who entertains people by doing magic tricks	0.075	0.09	0.03
A boy is a child who will grow up to be a man.	A rooster is an adult male chicken.	0.1075	0.2971	0.0795
A cushion is a fabric case lled with soft material, which you put on a seat to make it more comfortable	A jewel is a precious stone used to decorate valuable things that you wear, such as rings or necklaces.	0.0525	0.1745	0.1376
A monk is a member of a male religious community that is usually separated from the outside world.	A slave is someone who is the property of another person and has to work for that person.	0.045	0.1394	0.07
An asylum is a psychiatric hospital.	A cemetery is a place where dead peoples bodies or their ashes are buried.	0.375	0.03	0.02
A grin is a broad smile.	A lad is a young man or boy.	0.0125	0.028	0.024
The shores or shore of a sea, lake, or wide river is the land along the edge of it.	Woodland is land with a lot of trees.	0.0825	0.3192	0.1692
A monk is a member of a male religious community that is usually separated from the outside world.	In ancient times, an oracle was a priest or priestess who made statements about future events or about the truth.	0.1125	0.1	0.02
A boy is a child who will grow up to be a man.	A sage is a person who is regarded as being very wise.	0.0425	0.2305	0.038
An automobile is a car.	A cushion is a fabric case lled with soft material, which you put on a seat to make it more comfortable.	0.02	0.033	0.0167
A mound of something is a large rounded pile of it.	The shores or shore of a sea, lake, or wide river is the land along the edge of it.	0.0350	0.0386	0.044
Food is what people and animals eat.	A rooster is an adult male chicken.	0.055	0.2972	0.187
A cemetery is a place where dead peoples bodies or their ashes are buried.	Woodland is land with a lot of trees.	0.0375	0.124	0.0385
A forest is a large area where trees grow close together.	A graveyard is an area of land, sometimes near a church, where dead people are buried.	0.065	0.2787	0.1454
The shores or shore of a sea, lake, or wide river is the land along the edge of it.	A voyage is a long journey on a ship or in a spacecraft.	0.02	0.03	0.035
A bird is a creature with feathers and wings, females lay eggs, and most birds can y.	Woodland is land with a lot of trees.	0.0125	0.1334	0.018
A cemetery is a place where dead peoples bodies or their ashes are buried.	A mound of something is a large rounded pile of it.	0.0575	0.0842	0.041
A furnace is a container or enclosed space in which a very hot re is made, for example to melt metal, burn rubbish or produce steam.	An implement is a tool or other piece of equipment.	0.05	0.1408	0.0288
A crane is a large machine that moves heavy things by lifting them in the air	A rooster is an adult male chicken.	0.02	0.056	0.033
A car is a motor vehicle with room for a small number of passengers.	When you make a journey, you travel from one place to another.	0.0725	0.0261	0.016
Glass is a hard transparent substance that is used to make things such as windows and bottles.	A jewel is a precious stone used to decorate valuable things that you wear, such as rings or necklaces.	0.1075	0.2692	0.42
A magician is a person who entertains people by doing magic tricks.	In ancient times, an oracle was a priest or priestess who made statements about future events or about the truth.	0.13	0.1	0.026
A crane is a large machine that moves heavy things by lifting them in the air	An implement is a tool or other piece of equipment.	0.185	0.106	0.026
Your brother is a boy or a man who has the same parents as you.	A lad is a young man or boy.	0.1525	0.192	0.058
A sage is a person who is regarded as being very wise.	In legends and fairy stories, a wizard is a man who has magic powers.	0.2825	0.0452	0.038
In ancient times, an oracle was a priest or priestess who made statements about future events or about the truth.	A sage is a person who is regarded as being very wise.	0.035	0.166	0.0357
A bird is a creature with feathers and wings, females lay eggs, and most birds can y.	A crane is a large machine that moves heavy things by lifting them in the air.	0.1625	0.1704	0.0186

Bibliography

[1] Social media posts, videos and content uploaded survey. https://www.internetlivestats.com/one-second/ .

[2] Current commercial machine translation systems and computer-based translation tools: system types and their uses, John Hutchins, 2005. http://www.mt-archive.info/IJT-2005-Hutchins.pdf

[3] Luhn, H. P. (1958). The automatic creation of literature abstracts. IBM Journal of research and development

[4] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

[5] Wencan Luo, Fei Liu, Zitao Liu, and Diane Litman. 2016. Automatic summarization of student course feedback. In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL).

[6] Kai Hong, John M Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC).

[7] Joshua C. Feblowitz, Adam Wright, Hardeep Sing,Lipika Samal, Dean F. Sittig Summarization of clinical information: A conceptual model. Journal of Biomedical Informatics

[8] C. Fellbaum, WordNet. Wiley Online Library, 1998

[9] Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In Applied natural language processing.

[10] Jackie Chi Kit Cheung and Gerald Penn. 2014. Unsupervised sentence enhancement for automatic summarization. In Empirical Methods in Natural Language Processing.

[11] A. M. Rush, S. Chopra, and J. Weston, A neural attention model for abstractive sentence summarization, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 379{389.

[12]] S. Chopra, M. Auli, and A. M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93{98.

[13] R. Nallapati, B. Zhou, C. dos Santos, C . glar Gulc ehre, and B. Xiang, Abstractive text summarization using sequence-to-sequence RNNs and beyond, CoNLL 2016, p. 280, 2016

[14] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Neural Information Processing Systems.

[15] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In Association for the Advancement of Arti cial Intelligence.

[16] Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In Proceedings of NAACL-HLT

[17] Ziqiang Cao, Wenjie Li, Sujian Li, Furu Wei, Improving Multi-Document Summarization via Text Classi cation

[18] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, Dragomir Radev Graph-based Neural Multi-Document Summarization

[19] Kexin Liao, Logan Lebano , and Fei Liu. 2018. Abstract meaning representation for multi-document summarization. In Proceedings of the International Conference on Computational Linguistics (COLING).

[20] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111{3119

[21] Sanidhya Mangal LSTM vs. GRU vs. Bidirectional RNN for script generation

[22] H. Rubenstein and J. B. Goodenough, Contextual correlates of synonymy. Communications of the ACM, vol. 8, no. 10, pp. 627{ 633, 1965.

[23] The Unreasonable E ectiveness of Recurrent Neural Networks http://karpathy.github.io/2015/05/21/rnn-effectiveness/ , Andrej Karpathy blog

[24] Atish Pawar, Vijay Mago, Calculating the similarity between words and sentences using a lexical database and corpus statistics IEEE transactions on knowledge and data engineering 2018.

[25] Vitalii Zhelezniak, Aleksandar Savkov, April Shen Nils Y. Hammerla, Correlation Coe cients and Semantic Textual Similarity, Babylon Health.

[26] D. Bollegala, Y. Matsuo, and M. Ishizuka, Measuring semantic similarity between words using web search engines. www, vol. 7, pp. 757{766, 2007

[27] Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In Multi-source, Multilingual Information Extraction and Summarization, Springer, pages 3{21.

[28] Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In International ACM SIGIR conference on Research and development in information retrieval

[29] Chris D Paice. 1990. Constructing literature abstracts by computer: techniques and prospects. Information Processing Management 26(1):171{186.

[30] Ramiz Aliguliyev Multidocument summarization by sentence ranking, Second International Conference on Optimization with Industrial Applications

[31] Sherry, Parteek Bhatia Multilingual text summarization with UNL, 2015 International Conference on Advances in Computer Engineering and Applications

[32] Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, Chengqing Zong, Neural Cross-Lingual Summarization

[33] Qiwei Yin, Ruixun Zhang, and Xiu Li Shao CNN and RNN mixed model for image classi cation

[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan Show and Tell: A Neural Image Caption Generator

[35] Semantics de nition https://en.wikipedia.org/wiki/Semantics .

[36] Sepp Hochreiter, Jurgen Schmidhuber, LONG SHORT-TERM MEMORY

[37] Sepp Hochreiter Untersuchungen zu dynamischen neuronalen Netzen Diploma thesis, TU Munich, 1991

[38] Katja Filippova, Mihai Surdeanu, Massimiliano Ciaramita, Hugo Zaragoza, Company-Oriented Extractive Summarization of Financial News, Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)

[39] J. O'Shea, Z. Bandar, K. Crockett, and D. McLean, Pilot short text semantic similarity benchmark data set: Full listing and description Computing, 2008.

[40] Colah's blog: Understanding LSTM Networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[41] Wikipedia de nition for TFIDF vectorizers, https://en.wikipedia.org/wiki/Tf\OT1\textendashidf

[42] Atish Pawar and Vijay Mago, Calculating the similarity between words and sentences using a lexical database and corpus statistics, IEEE transactions on knowledge and data engineering, February 2018.

[43] NIST organization: https://www-nlpir.nist.gov/projects/duc/index.html .

[44] Bleu automatic evaluation metric, https://en.wikipedia.org/wiki/BLEU .

[45] Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. IBM Research Report RC22176 (W0109-022).

[46] Meteor automatic evaluation metric, https://en.wikipedia.org/wiki/METEOR.

[47] ROUGE: A Package for Automatic Evaluation of Summaries. https://www.aclweb.org/anthology/W04-1013 .Chin-Yew Lin, Information Sciences Institute University of Southern California,

[48] Cormen, T. R., C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. The MIT Press 1989

[49] Logan Lebano , Kaiqiang Song and Fei Liu. https://arxiv.org/pdf/1808.06218.pdf Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization.

[50] WikiNews site: http://www.wikinews.org/

[51] Github repository with reference summaries and article topics: https://github.com/NightFury13/TextSummarizer/tree/master/duc-2004?fbclid=IwAR0BTdBumh4G_0kOp7rqR3Fme1Hs6zbQeJ0UsGfsksSdLCsow9rq7rJ8MIw

[52] Abigail See, Peter J. Liu, Christopher D. Manning https://arxiv.org/pdf/1704.04368.pdf. Get To The Point: Summarization with Pointer-Generator Networks

[53] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy. Neural Abstractive Text Summarization with Sequence-to-Sequence Models, December 2018