POLITECNICO DI TORINO

Master degree course in Computer engineering
Software

Master Degree Thesis

WEB-ELSA: Integrating word embeddings into itemset-based summarizer

Supervisors
Luca Cagliero
Paolo Garza

Candidate
Gianluca Di Giacomo

Academic year 2019/2020

Summary

The use of electronic textual documents is gaining importance in many and different environments. For example, the majority of educational resources is often shared through e-learning platforms. In many other contexts, the digitalization is increasing: one need only think of news retrieved from online newspaper or social media. Therefore the need for summarization techniques to extract information in short time is consistently raising. Many attempts have been done with different techniques and models. This thesis work aims at analyzing the impact of word embedding information to improve the performances of an itemset-based summarizer.
Itemset-based summarization is among of the most promising techniques to compute summaries on multi-document collections. It is composed by three phases: the first is the text transformation and modelling, in which the texts are transformed in an useful format for the algorithm, the so-called transactional dataset. The second phase is the frequent itemset extraction, which consists of finding the recurrent co-occurrences of items in the transactional dataset, that can be done with different techniques. The last step is the sentence selection: the most valuable original text sentences, chosen by the algorithm through the frequent itemset mining, are extracted to be part of the final summary. This thesis work has an important impact on the first and third phases, while the second is unchanged. This thesis work aims at extendingn ELSA [1], which is a summarization algorithm based on Latent Semantic Analysis and itemset extraction. It is the most recent work about itemset-based summarizers: other examples are ITEMSUM [19], GRAPHSUM [2] and MWI-SUM [3]. ELSA, which stands for Enhanced Latent Semantic Analysis, has been developed to exploits the best part of the two techniques: the itemset ability to consider correlations among multiple words and the LSA ability to synthetize textual content into meaningful concepts. The concepts extracted through LSA are, originally, represented by a set of singular terms, while in ELSA the concepts are represented as a set of itemsets, that can be composed by one or more words. This allows to better

characterize a concept with more information. It is also worth noticing that ELSA was developed to be easily portable to any language, so it does not depend on the syntactic and semantic structure of a particular language.
Word embedding is a set of techniques to give word a continuous representation, that means associating a vector of real numbers to a word, such that vectors for syntactically and semantically similar words are correlated. The vectors representing words can be computed with different models on large unlabeled corpora. To the best of our knowledge, the most used techniques are based on corpus statistics or neural networks. A pioneering work in this field was word2vec [4], that is a clear example of the usage of Neural network to obtain embeddings. Another important work about embeddings is GloVe [5] that, unlike word2vec, uses corpus statistics to compute embeddings. For this thesis, the pre-computed vectors of FastText [7] are used because of a peculiar feature of the FastText algorithm: the possibility to create embeddings for unpresent words in the original training corpus. The corpora used for training the vectors are those related to the work Learning Word Vectors for 157 Languages[15]. Vectors are computed through a model that exploits also the morphological information in words, where each word is represented as a bag of character n-gram. From this point on, we will refer to the modified variants of ELSA as WEBELSA (Word Embedding-Based ELSA). Three variants of the WEBELSA algorithm are explored in this thesis and tested on the Multiling'13 pilots, that is an international competition through which different algorithms can be compared under the performance point of view, on different languages. The tests were made on 8 languages: Arabic, Czech, English, French, Greek, Hindi, Romanian and Spanish. How it will be shown in detail in the next chapters, the performances of the summarizers are evaluated through an automatic tool that compare the generated summaries with the summaries that are generated by humans. The results obtained through this work are promising: it is clear that adding word embedding information to ELSA summarizer leads to an improvement both on tuned variants of ELSA and on standard ones for two of the three tested variants. Standard variants are obtained through a common parameters setting among all languages, while tuned variants are obtained exploring particular parameter combination for each language.
Finally, this work opens a great number of researches to explore more in the details the possible combination of itemset-based summarizers and embeddings in general. New word embedding techniques and pre-computed word vectors are now available. Moreover, different type of measures could be

exploited to better understand how much words are syntactically and semantically each other related. Furthermore, it would be very interesting to investigate on the possible usage of sentence embedding, that is a technique to associate a vector to an entire sentence: for example BERT [6] is among the most promising models in this field. Since summarization strategies are sentence-based, through sentence embedding, it would be possible have better comparisons among sentences instead of using single words.

Contents

Chapter 1

Related works

1.1 Itemset-based summarization

Recently, the attention of researchers was shifted on the multi-lingual document summarization techniques based on itemset mining and Information Retrieval. Itemset mining is an well-known data-mining technique that has largely been used to find out repetitive correlations between items in transactional datasets [16](Han et al. 2007). It is easy to think that combinations of words that often co-occur in different sentences, are probably words that describe important concepts that should be mentioned in the final summary, and in this way the itemset-based summarizers, in the sentence selection, are driven by this mining process. An initial attempt to consider word co-occurrences in text summarization was made in Lin and Hovy (2000)[17] but this approach is not scalable to large document collections. An abstractive method based on itemsets has been presented in Hynek and Jezek (2003)[18], while (Baralis et al. 2011 [20], 2012 [19], 2015[21]) propose extractive itemset-based approaches. The latter strategies is composed by the following steps: the mining of itemsets is done for first, then the sentences that contain the biggest number of frequent itemsets are chosen. In (Baralis et al. 2012, 2011) the algorithm extract itemsets using an entropy-based approach (Mampaey et al. 2011) [47] to limit the overlapping between the mined itemsets. In Baralis et al. (2015), itemset occurrences within sentences are weighted by a variant of the tf-idf measure, called tf-df (term frequency-document frequency). These techniques have two important drawbacks: the first is that the mined itemsets are intrinsically redundant because of the possible repetition of the same word in different itemsets and the second is that there is not a clear mapping between the mined frequent itemsets and the document concepts.

1.2 Latent semantic analysis for summarization purposes

Another branch of research was focused on applying the Latent Semantic Analysis for multi-document summarization . LSA-based summarizers (e.g., Steinberger et al. (2011) [22], Ozsoy et al. (2011) [23], and Steinberger and Jezek (2009) [24]) apply SVD technique, that stands for Singular Value Decomposition, to a term-by-sentence matrix, where on the rows of the matrix there are the text words, while on the columns there are the sentences of the text and in the elements of the matrix the occurrences of single terms within the sentences of the text are stored. The matrices obtained by applying the SVD technique can be used to produce a concept-by-sentence matrix, where a concept is represented by a set of words that occur in similar part of the text. Sentences with largest coverage of the LSA-based concepts are selected to produce the final summary. In Steinberger and Jezek (2009) single word occurrences in the term-by-sentence matrix are weighted by an entropy-based meter, while Steinberger et al. (2011) combined LSA-based summarization with named entity recognition and disambiguation. The JRC summarizer proposed in Steinberger et al. (2011) reached top results in the MultiLing Pilot of the TAC'11 contest (Giannakopoulos et al. 2011) [26]. An experimental comparison between different LSA-based summarizers is given in Ozsoy et al. (2011). The main limit of LSA-based strategies is the inability to correlate sets of terms with the underlying document concepts.

Chapter 2

General introduction to word embedding and FastText model

2.1 General models

Word-embedding is a set of techniques to give words a continuous representation, i.e. a vector of reals is associated to a given word. There are many techinque to accomplish this task: for example, it is possible to use co-occurrence-statistics or RNNs (Recurrent Neural Networks). The GloVe model computes the word vectors starting from a word-word matrix, that counts the co-occurrences of a word in the context of another word, also through probe words. The dimensions of the matrix are huge, so a dimensionality reduction work is then done. The vectors used for this thesis work are those computed by the FastText model, that is based on the [14] Mikolov et al. (2013b) model. In that model the words representation in vector space was studied, and the syntactic and semantic relationships amongs words were represented as offset vectors. One of the most famous examples was the one about the male/female relationship: subtracting from the vector associated to the word "king" the vector of the word "man" and adding the vector associated to the word "woman", a vector very close the the one of the word "queen" is obtained. The very interesting part of this model is that these relationships are learned in an unsupervised way, showing the power of Recurrent Neural Network models. Given a large training corpus represented as a sequence of words $w_1, \ldots, w_T$, the objective of the skipgram model with negative sampling 2.1 is to maximize the following log-likelihood:

$$\sum_{t=1}^{T} \sum_{c \subseteq C_t} \log p(\ w_c \mid w_t\ )$$

where the context C_t is the set of indices of words surrounding word w_t. Simplifying the definition, the objective of the skipgram model is, given a word w_c, predict the words that appear in the context of w_c. For readability,

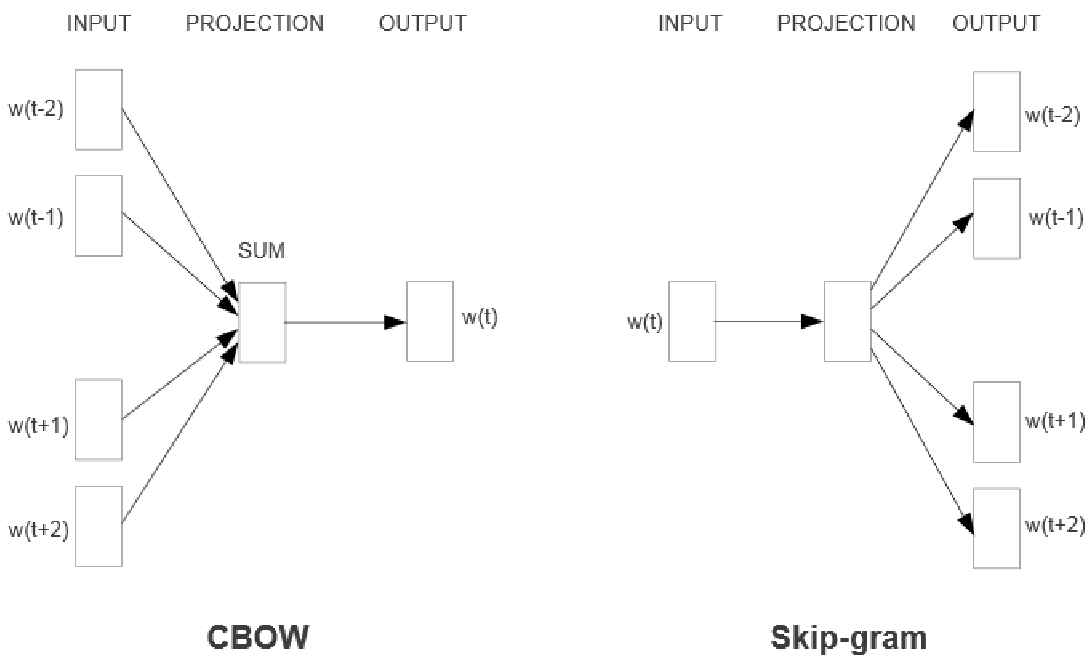

Figure 2.1: General architecture of CBOW and Skipgram models (Picture taken from "Efficient Estimation of Word Representations in Vector Space" [31]).The Cbow architecture predicts the current words from the context ones, while the skipgram model predicts the context words from the chosen one. During the training the word position is not considered, the context is treated as a bag of words.

further mathematical steps are not reported here but they are available in the Mikolov et al. (2013b) [14].

2.2 FastText model

The innovation of FastText is to model the morphology of the words, considering them as a bag of subwords. For example , considering subwords of 3 characters, the word "house" can be decomposed in

$$< ho, hou, ous, use, se >$$

At this group the word itself "house" need to be added. Assuming a dictionary of subwords of size D is given, and $D_w \subset \{1, \ldots, D\}$ the set of subwords appearing in a word w is denoted: associating a vector representation s_d to each subword d, the scoring function that allows to take into account the morphology of the word is obtained as

$$s(w, c) = \sum_{d \in D_w} s_d^T \cdot v_c$$

In this way, words that have similar morphology do not have completely distinct vectors. Furthermore, considering a word-vector as a sum of subwords vectors appearing in that word, vectors for unknown words can be computed. This can be very useful to compute vectors for rare words. For this reason, FastText 300 dimensional vectors were chosen among all the possible models.

2.3 An overlook of the existing embedding-based summarization method

Following the work of Lucas de Haas [8] , a briefly description of the most used methods that exploit embedding information is now given. The first method is the one proposed by Lin and Bilmes(2010)[9] in which a greedy algorithm is exploited to optimize two submodular objective functions where the cosine similarity between the TF-IDF vectors of two sentences is used. In 2011, Lin and Bilmes introduced a combination of two functions that estimate the coverage and diversity exploiting cosine similarity. Kågebäck et al. (2014) [11] do not use embeddings based on tf-idf measure, but sentence embeddings that are built through word embeddings of the sentence, put together in an unsupervised way by an unfolding recursive auto-encoder, based on Socher et al.(2011)[12]. In the same framework of Lin and Bilmes, Kobayashiet al.(2015) [13] proposed two totally different objective functions, called DocEmb and EmbDist. The main differences between these algorithms and WEBELSA variants are two: WEBELSA variants always use embeddings not based on TF-IDF measure and the WEBELSA sentence embeddings are computed in a peculiar way 3.6.

Chapter 3

Introduction to ELSA and WEBELSA

3.1 The Enhanced Latent Semantic Analysis model

ELSA is the core part of this thesis work. It is a multilingual document summarization algorithm based on frequent itemesets and Latent Semantic Analysis. It has been developed to be language independent and it represents the state of the art of itemset-based summarizers. It combines the ability of Latent Semantic Analysis to synthetize into concepts the content of large and homogeneus document collections with the ability of itemset-based models in representing documents through recurrent combinations of multiple words. In the ELSA summarization model concepts are described by a set of frequent itemsets, which may consist either of single terms or of sets of terms of arbitrary size. For this thesis work, we choose a concept to be represented by 15 itemsets.

3.2 Structure of the algorithm

The algorithm can be divided in four parts:

1. **Document preparation.** Documents are marginally elaborated to be used in the step 2, 3 and 4 of ELSA.

2. **Frequent itemset mining.** Frequent itemsets are extracted from

transactional dataset. The frequent itemsets are then attached to the original sentences to produce an itemset-by-sentence matrix.

3. **Singular Value Decomposition.** The itemset-by-sentence matrix is decomposed using Singular Value Decomposition (SVD) to produce a concept-by-sentence matrix, where a concept can be expressed by multiple frequent itemsets.

4. **Sentence selection.** The sentences covering the most important relevant concepts are chosen. To avoid redundancy, a sentence is chosen if it is not too similar to the previous chosen ones.

Figure 3.1 is a graphical representation of all main steps of WEBELSA algorithm.

3.3 Major modifications

The key idea to improve the performances of ELSA through word embedding is to use the cosine similarity among word vectors to perform computations. FastText creates embeddings so that the cosine similarity among vectors of syntactically or semantically related terms is high. This is a fundamental aspect because gives information about word semantic, that is exploited in two different points of ELSA:

- in step two, for the itemset-by-sentence matrix construction;
- in step four, for the sentence redundancy value computation.

3.4 Document preparation

The document preparation has the following simple steps, available for most of spoken languages and two important parameters are introduced:

1. **Stopword elimination.** The stopword elimination attempts to removing all words that have negligible lexical meaning (e.g. articles, prepositions,conjunctions) even if they have a great number of occurrences.

2. **Stemming.** In this step, words occurring in sentences are reduced to their base form, called *stem*. This improves the quality of frequency-based term analysis.

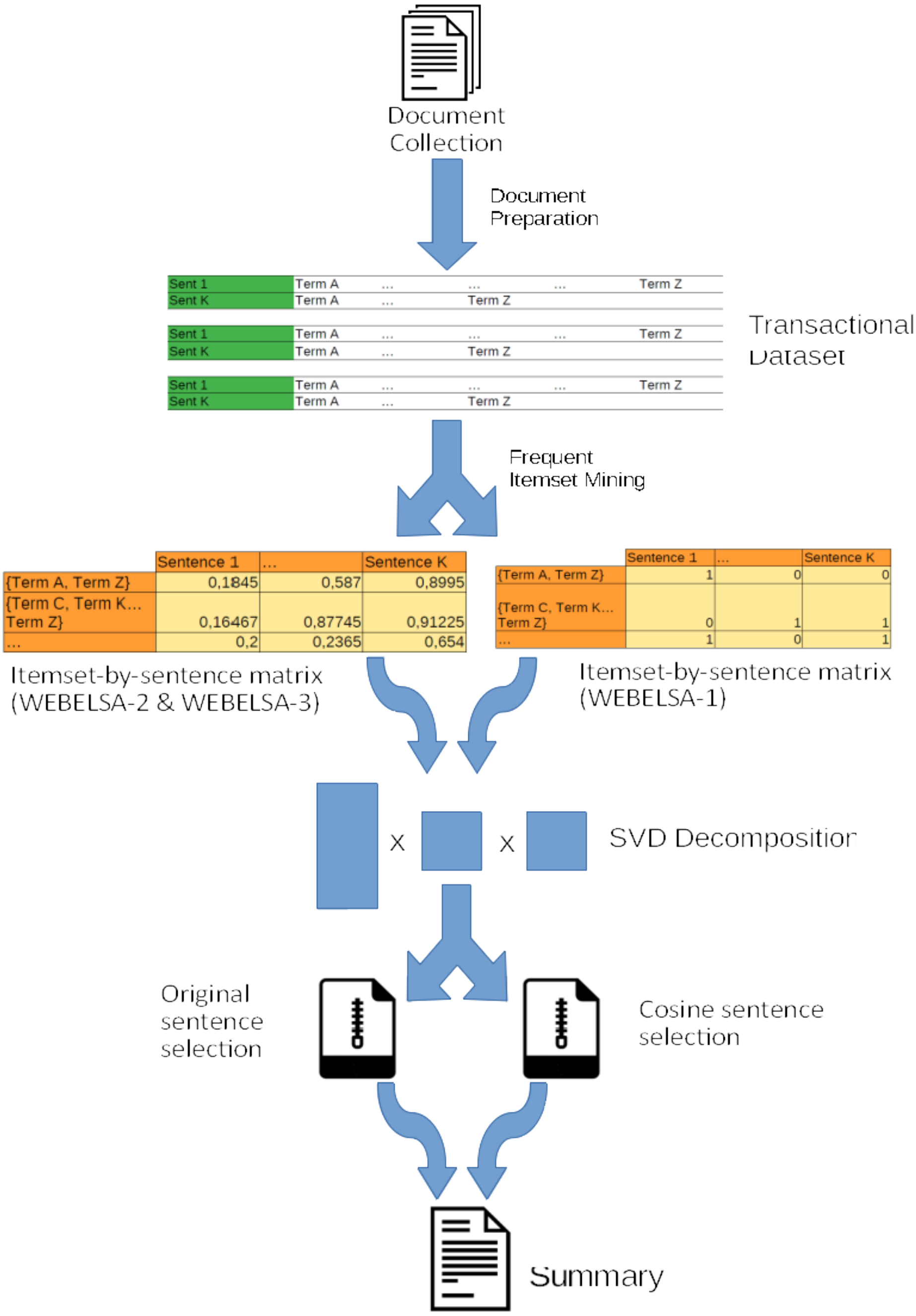

Figure 3.1: General structure of the algorithm.

3. **Sentence pruning.** Due to the fact that we are dealing with news articles, it is easy to imagine that the last sentences, for position in texts, are partly repeated. Thus, only the first *numSentences* sentences of the articles are elaborated. Furthermore, to increase the quality, the short sentences that share more than a percentage of stems, denoted as $overlapThreshold$, are removed.

4. **Transactional dataset.** The last step of preparation is documents transformation in transactional data format, for the following itemset mining phase.

In few words, while doing the transformation of the document collection in the transactional dataset, two steps are performed: all documents are merged in a single dataset and each sentence is transformed in a transaction consisting of a set of stems.

3.5 Frequent itemset mining and itemset-by-sentence matrix construction

The first part of this section is about finding the set of frequent itemsets occurring in the transactional dataset. In this context, an itemset is composed by a set of distinct stems that co-occur in any transaction of the transactional dataset. $Frequent\ itemsets$ are itemsets whose support value exceed an analyst-provided (named in this thesis *minSupportPercentage*) threshold. For the finding of frequent itemsets, the FP-Growth [25] algorithm was used.
The second part of this section is about building an itemset-by-sentence (IS) matrix that will be exploited in the next step. The IS is build in the following way: on the matrix rows the frequent itemsets are placed while on the matrix columns the sentences are positioned. Matrix cells can contain only two possible values: 0 or 1. A matrix cell c_{ij} has the value 1 if the i-th frequent itemset is contained in the j-th sentence, 0 if the itemset is not contained.

3.6 WEBELSA-ItemsetWeights and WEBELSA-Redundancy+Weights: itemset-by-sentence matrix construction

How seen in the second part of 3.5, the itemset-by-sentence matrix in ELSA is built, inserting the value 1 in the matrix cell c_{ij}, corresponding to the i-th frequent itemset and the j-th sentence, if the i-th itemset is contained in the j-th sentence and the value 0 if it is not contained. In WEBELSA-ItemsetWeights and WEBELSA-Redundancy+Weights, the cosine similarity measure between the *itemset embedding* vector and the *sentence embedding* vector is inserted in all matrix elements:

$$c_{ij} = \cos\theta = \frac{\sum_{z=1}^{z=\alpha} a_z b_z}{\sqrt{\sum_{z=1}^{z=\alpha} a_z^2}\sqrt{\sum_{z=1}^{z=\alpha} b_z^2}}$$

where α is the vector dimensionality. In this thesis work, α is equal to 300. The *itemset embedding* vector **A** is computed as the average, on each dimension, of word embeddings of the words contained in the itemset.
The *sentence embedding* vector **B** is computed as the average, on each dimension, of the word embeddings of words contained in the sentence.
The goal of using cosine similarity is to add information in the IS matrix, to specify the relationship between itemsets and sentences: an itemset could be not contained in a sentence, but still semantically related to that.

3.7 Singular Value Decomposition

This step reduces the itemset-by-Sentence matrix to a Concept-by-Sentence (CS) matrix thorough Singular Value Decomposition (SVD). After SVD reduction, each frequent itemset is mapped to a subset of latent concepts identified in the texts collection.
Suppose you are given a $|I|$ x $|T|$ matrix **IS** matrix of rank r, we can assume $|I| \gg |T|$ without losing generality, (Steinberg and Jezek, 2005 [44]), the SVD matrix is defined as

$$\mathbf{IS} = U \cdot \Sigma \cdot V^T$$

where

- -**U**: $|I|$ x r orthonormal matrix ($U^{-1} = U^T$) whose columns are usually denoted as left singular vectors.

- -Σ: r x r diagonal matrix, whose diagonal elements are positive singular values
- -**V**: r x $|T|$ orthonormal matrix, whose columns are usually denoted as right singular vectors.

The diagonal matrix Σ has r positive elements $\sigma_1 \ldots, \sigma_r$ that correspond to r different concepts, in a decreasing order. Only the highest $k < r$ values, that correspond to the top-k concepts, are taken into account. The number k is chosen in the following way: all σ whose value is higher than a given threhsold are chosen. The threshold value corresponds to the half of the highest σ value. In the matrix V^T only the first k rows are important: they represent the concepts produced by SVD. On the columns of V^T there are the sentences of the document collection. Each element (in the first k rows) shows the importance of a sentence with respect to a concept. Thanks to the SVD techique, the selection of the first k singular values reduces the frequent itemset space I to a k dimensional singular vector space. Itemsets related to the same concepts are mapped to the same region in the transformed space. The matrix produced from the operation $U \cdot \Sigma^k$ shows the importance of an itemset for a concept. The top-k concepts are placed on the columns of the matrix, while the frequent itemsets are placed on the rows. Each element of the matrix represents the relationship of a given itemset with a concept. This allows to exploit the itemsets representation of concepts instead of classic single-word representation of concepts. Only the first 15 itemsets linked to a concept are picked.

3.8 Sentence selection

The final step of ELSA and WEBELSA summarizers is the final sentence selection. To improve the quality of the final summary, two measures with two different purpose are used: the sentence significance and the sentence redundancy. The first measure is used to estimate how much an original sentence coverages the most important concepts found by the SVD technique. The second measure estimates the redundancy of a sentence with respect to the previously extrapolated ones. Our intent is to maximize the sentence significance, while minimizing the sentence redundancy.

Sentence significance. To evaluate the sentence significance based on the SVD, two values are used: the σ_q of the diagonal matrix, that gives information about the importance of a concept and the value v_{qx} of the matrix V^T that measures the relationship between the sentence x and the concept q. The sentence significance is:

$$sign(s_x) = \sqrt{\sum_{q=1}^{k} \sigma_q^2 \cdot v_{qx}^2}$$

Sentence redundancy. The concept-by-sentence matrix is computed as $\mathbf{F} = \Sigma^k \cdot V^T$, where each element contains the value $f_{qx} = \sigma_q \cdot v_{qx}$. A comparison of the pairs coverage weights associated with the same singular value is done, to compare the redundancy of two sentences. If all values are concordant and similar, their coverage of LSA concept is similar. The sentence redundancy is:

$$sim(s_x, s_y) = \begin{cases} 1 & \text{if } \frac{min(f_{qx}, f_{qy})}{max(f_{qx}, f_{qy})} \geq similarityThreshold \ \forall \ q \in [1, k] \\ 0 & otherwise \end{cases}$$

where *similarityThreshold* is an analyst-set parameter. A candidate sentence is discarded if the similarity with any of the sentences already in the summary is equal to 1. The sentences are ordered by significance, in this way new sentences which are too similar to those previously inserted in the output summary are discarded.

3.9 WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights: sentence redundancy

In the original ELSA algorithm, the final sentence redundancy computation 3.8 is done among columns of the matrix $F = \Sigma \cdot V^T$. In WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights, the choice of including a sentence in the output summary is done by using cosine similarity of the *sentence embeddings*.

Given the set of sentences $S = \{s_1, \ldots, s_n\}$ already included in the output summary and their respective embeddings $emb_1, \ldots, emb_n$ and given a candidate sentence s_y to include in the summary, and its respective sentence embeddings emb_y, the sentence similarity value is:

$$sim(s_x, s_y) = \begin{cases} 1 & if \ cosineSimilarity(emb_k, \ emb_y \ ) \geq cosineThreshold \ \ \forall \ k \in [1, n] \\ 0 & otherwise \end{cases}$$

where *cosineThreshold* is an analyst-provided parameter. As for the original version, a candidate sentence is discarded if the similarity with any of the sentences already in the summary is equal to 1. Even in this case, the sentences are ordered by significance. This modification aims at better exploiting the semantic of whole sentences in the final sentence selection phase.

Chapter 4

Experiments

4.1 Document collections

All three WEBELSA algorithm variants were tested, like ELSA, on the MultiLing'13 pilot (Giannakopoulos 2013 [28]), that is a multi-language summarization task proposed in the Multiling workshop at ACL 2013 conference. The original document collection was translated from the English language to other 10 languages: Arabic, Chinese, Czech, English, French, Greek, Hebrew, Hindi, Romanian, and Spanish. The initial English collection was made by 15 document groups, about different topics, and each group was made by 10 different documents, for a total of 150 documents per language, with the exception of the French, Hebrew and Hindi languages for which just 10 different topics (and so 100 documents) were available. A golden summary (i.e., the optimal document collection summary) was created for every group for the English language, and then translated in all languages. To estimate the quality of the generated summaries, a comparison is done between the generated summary and the golden ones generated by humans, through an automatic toolkit called JROUGE (Krapivin et al. 2014 [29]). It measures the overlap between the compared summaries. The metrics used to assess the quality of the generated summary were ROUGE-4, which is among the most accurate, as reported in Rankel et al. (2012) [30], and ROUGE-2. To perform a clear comparison, the generated summaries were truncated at a length of 250 words as recommended by the MultiLing pilot organizers (Giannakopoulos 2013; Giannakopoulos et al. 2011 [26]). All three variants of the algorithm were tested on 8 languages: Arabic, Czech, English, French, Greek, Hindi, Romanian and Spanish.

4.2 WEBELSA parameters

Five parameters are analyzed:

1. sentence overlap threshold ("overlapThreshold");
2. minimum support ("minSupportPercentage");
3. cosine threshold ("cosineThreshold");
4. sentence similarity threshold ("similarityThreshold");
5. number of initial sentences ("numSentences").

The sentence overlap, the minimum support and the number of initial sentences are used and modified in all three WEBELSA variants, the cosine similarity in WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights variants and the sentence similarity only in WEBELSA-ItemsetWeights version. The sentence overlap is the parameter that is used for early pruning sentences that are too similar. The minimum support is the parameter used for the frequent itemset mining. The number of initial sentences is the number of sentences, starting from the first, that are considered while reading the news. The cosine similarity is the parameter used for the final sentence selection in WEBELSA-RS and WEBELSA-RS+IW. The sentence similarity is the parameter used for the final sentence selection in WEBELSA-IW.

4.2.1 Parameters value for standard variants of WEBELSA

We empirically studied the impact of the parameters on the results of the summarizer. For WEBELSA-RS, the standard configuration is:

- overlap threshold = 50%
- minimum support = 6%
- number of initial sentences = 5
- cosine threshold= 0.95

For WEBELSA-IW, the standard configuration is:

- overlap threshold = 50%
- minimum support = 7%

- number of initial sentences = 5
- sentence similarity threshold= 99.9%

For WEBELSA-RS+IW, the standard configuration is:

- overlap threshold = 50%
- minimum support = 2%
- number of initial sentences = 5
- cosine threshold= 0.925

For the sentence overlap threshold, the best value is 50%: a higher value could lead to the loss of positive early pruning effect of close sentences. On the other hand, a lower value could be too restrictive and would cause the loss of sentences with partial new information. The range recommended for this parameter is between 45% and 60%. It is also worth noticing that there experiments which prove that values outside this range could improve performances. The minimum support affects the mining of frequent itemsets: a too high value decreases the number of frequent itemsets, reducing the dimensions of the itemset-by-sentence matrix. A low value exponentially increases the quantity of frequent itemsets, and this would increase overlap among mined itemsets. On average, this parameter should be chosen in the range 2%-7%. An high number of initial sentences can cause the repetition of information, thus leading a decrease of performances of the summarizer. Instead, a low number could lead to an important information impoverishment. How it is possible to see, the best value of initial sentences to read is 5 for standard variants of the algorithm, but a good range to explore is 3-7. The cosine threshold and the similarity threshold are the parameters used for the final sentence redundancy computation. An high value for these parameters could be too permissive, such that sentences that bring few new information would be inserted in the output summary, causing redundancy. On the other hand, a too low value would cause loss of information, reducing the length of the summary. For the sentence similarity threshold, only very high values lead to satisfying results: 99.9% and 100%. By setting lower values the algorithm performance decreased. For the cosine similarity, a good range to use is between 0.90 and 0.95.

4.3 Algorithm execution time

ELSA generally takes few seconds or minutes to complete all the computations for all considered languages. This is not true for the WEBELSA variants. The major reason for long times of execution, in the order of tens of minutes for a single language , has to be searched in the research phase of word embedding in the files that contain the associations word - embedding. Another reason for long execution time is the number of computations that the algorithm needs to perform: the itemset-by-sentence matrix can have milions of frequent itemsets, thus milions of rows. This can leads to tens of millions of computations for a single matrix, that represents a single collection of a language. The execution time is generally longer for WEBELSA-IW and WEBELSA-RS+IW implementations, where the cosine similarity among frequent itemsets and sentences is computed, while for WEBELSA-RS+IW these calculations are made among few tens of sentences.

4.4 Competitors

A comparison of all three variants of WEBELSA summarizer was made with the following competitors that were submitted to the MultiLing'11 and MultiLing'13 contests.

1. the JRC summarizer (Steinberger et al. 2011; Steinberger 2013 [43]), which is the most efficient LSA-based summarizer.
2. the MWI-Sum, which is the most efficient itemset-based summarizer.
3. the Association Mixture Text Summarization (AMTS) (Gross et al. 2014).
4. an ILP-based summarizer, the ICSISumm multiple-document summarization system (ICSIsumm) (Gillick et al. 2009 [32], 2008 [33]).
5. the SubModular [Lin and Bilmes 2010].
6. Another set of summarizers that include: LexPageRank (Radev et al. 2004 [34]), Wan et al. 2007 [35], Mihalcea and Tarau 2004 [36], Erkan and Radev 2004 [37], Gillick et al. 2009 [38], Lin and Bilmes 2012 [39], Wan and Yang 2008 [40].
7. the Open Text Summarizer (OTS) (Rotem 2011 [41]) .

8. TexLexAn (TexLexAn 2011 [42]).

4.5 Summary examples.

The following are the examples produced by algorithms:

See next page

Algorithm	**Summary**
ELSA	Several prominent Romanian artists and celebrities will gather at the Radio Hall in Bucharest on Sunday, January 23 to raise money for the victims of the catastrophic Indian Ocean tsunami of December 26, 2004. Tuesday, January 18, 2005, Through a telethon, Romanians have raised 15 billion lei, or the equivalent of 395,000 euro for victims of the devastating Indian Ocean tsunami of Boxing Day 2004 that claimed more than 175,000 victims. Saturday, March 26, 2005 Up to four times as many women as men died in the December 26 Indian Ocean Tsunami, figures published by Oxfam International today reveal. In an event titled Romanian Artists in Support of Asia, organised by The Reporter Foundation of Romania, artists will auction off their works, as well as personal objects, with all proceeds being donated to the relief efforts for the tsunami victims. U.S. citizens donating in 2005 to help tsunami victims may write off their donations on their 2004 tax returns, thanks to a bill quickly passed in the U.S. House of Representatives and the U.S. Senate on a voice vote, and signed into law by president George W. Bush. According to new information about the earthquake of December 26, 2004, it was the longest-lasting earthquake ever recorded. A spokesman for the Red Cross speculated that the toll could increase to over 100,000 as some of the smaller islands in the Indian Ocean are checked, and a U.N. official said that the death toll might eventually approach 80,000 in Indonesia alone.

See next page

WEBELSA-IW	Tuesday, January 18, 2005Through a telethon, Romanians have raised 15 billion lei, or the equivalent of 395,000 euro for victims of the devastating Indian Ocean tsunami of Boxing Day 2004 that claimed more than 175,000 victims. The immediate fatalities from the earthquake and resulting tsunamis are but a fraction of the total effect from the disaster. In an abrupt about-face, the world's wealthiest nations have begun pouring funding into the Earthquake/Tsunami damaged region. Several prominent Romanian artists and celebrities will gather at the Radio Hall in Bucharest on Sunday, January 23, to raise money for the victims of the catastrophic Indian Ocean tsunami of December 26, 2004. Saturday, March 26, 2005 Up to four times as many women as men died in the December 26 Indian Ocean Tsunami, figures published by Oxfam International today reveal. The death toll from the earthquake and tsunamis that has hit countries in Asia and Eastern Africa continues to rise, passing 80,000 people according to reports from several news agencies. Gender imbalance in Tsunami deaths Nearly a week after tsunami waves scoured the coasts of multiple countries in southern Asia, the confirmed death count is over 120,000. Saskatchewan town asks for return of "accidental" tsunami donation In an event titled Romanian Artists in Support of Asia, organised by The Reporter Foundation of Romania, artists will auction off their works, as well as personal objects, with all proceeds being donated to the relief efforts for the tsunami victims. Egeland later recanted his statement, adding that America's contributions to Asia's tsunami relief was "one of the most generous pledges so far." According to new information about the earthquake of December 26, 2004, it was the longest-lasting earthquake ever recorded.

See next page

A spokesman for the Red Cross speculated that the toll could increase to over 100,000 as some of the smaller islands in the Indian Ocean are checked, and a U.N. official said that the death toll might eventually approach 80,000 in Indonesia alone. 2004 Sumatra quake was longest ever recorded Meadow Lake, Saskatchewan – The municipal council of a small town in Canada's Prairies has said it "accidentally" donated $10,000 to the Red Cross for tsunami relief. Indonesia reports it is no longer counting bodies, but merely struggling to deal with the aftermath and prevent a massive outbreak of typhoid. On Monday of this week, Jan Egeland, the UN's chief of emergency relief said that rich nations like the U.S. were being "stingy" by making small contributions. The ten-hour telethon was organised on Sunday, January 16, by the television station Realitatea TV, and was the first time an event of this kind had been staged in the country. While complaints about the 'miserly' generosity of the Bush Administration have surfaced in recent days, donations and actions at the grassroots level have quietly illustrated the concern and sympathy felt by ordinary Americans. "Normally, a small earthquake might last less than a second; a moderate sized earthquake might last a few seconds. Promised funds have doubled in the past 24 hours, to nearly 2 Billion U.S. dollars (USD). This disproportionate impact will lead to problems for years to come unless everyone working on the aid effort addresses the issue now. Aid pledges rise; Japan promises 500,000,000 USD We are already hearing about rapes, harassment and forced early marriages. Japan tops the U.S. Media celebrities such as the soprano Felicia Filip and singers Angela Similea and Dida Dragan will also be participating, as well as Maia Morgenstern, known internationally for her role in the film The Passion of the Christ. The council approved the donation during a Jan.

End

4.6 Achieved scores and general statistics

In the following table 4.2, the Borda Count re-ranking strategy is applied (van Erp and Schomaker 2000 [45]). It is a consensus function to compute an unique ranking of all tested summarizers, over multiple single rankings, one for each language. Here only the summarizers that have been tested on all languages are considered: for each language, an increasing integer is assigned to each summarizer, in decreasing order of ROUGE-4 and, separately, ROUGE-2 score. At the end, an average of these partial rankings is computed, one for ROUGE-4 and one for ROUGE-2, where a summarizer with a lower average is better than a summarizer with an higher average. The ranking shows that, for tuned variants, two algorithms, WEBELSA-RS-TUNED and WEBELSA-IW-TUNED, are better than Tuned ELSA for the ROUGE-4 measure, meanwhile only WEBELSA-RS-TUNED is better than Tuned ELSA for measure ROUGE-2. The WEBELSA-RS+IW-TUNED has a lower ranking than Tuned ELSA on both measures. For standard variants, WEBELSA-RS and WEBELSA-IW have a better ranking than ELSA. We can say that there are at least two methods that perform better than ELSA. Moreover, from the tables it appears that WEBELSA-RS performs better than ELSA on 5 languages, with the exceptions of Spanish (on which they have almost the same score), Romanian and Hindi. Moreover, WEBELSA-IW and WEBELSA-RS+IW have better performances than the ELSA ones on 6 languages, with the exceptions on Hindi and Spanish languages. This proves that the use of word embeddings can improve the performances, in a solid way, of an itemset-based summarizer. Unfortunately, not all parameter configurations could be tested due to the huge dimensions of the itemset-by-sentence matrix. In fact, by setting low or mean values for the minimum support parameter, the number of rows exponentially increases, causing the failure of the computations. This is true above all for Romanian language, but also for some Arabic and Spanish language configurations: maybe some of these missing configurations could further improve the performances, even if the experiments performed with success have acceptable results. From the summary examples, it is possible to see how the position of the sentences changes in the summaries. Even if the sentence position does not influence the scores computed by the toolkit, for human readability it is important: in the summary provided by ELSA summarizer, the first sentence is about the fundraising for an Ocean tsunami but there are not details about the raised funds quantity or about the human cost of the tragedy. Instead, in the summary provided by WEBELSA-IW summarizer, the first sentence

gives information about the quantity of the funds raised and also about the victims number of the tsunami. The same sentence is present also in the summary provided by ELSA, but in second position. To the best of our comprehension, it seems that, comparing the two summaries, more detailed information is given in the initial sentences by WEBELSA-IW summarizer. The Hindi language is the only language on which only the ROUGE-4 measures of WEBELSA-*-TUNED variants are satisfying, while the other scores are very low compared to the others. The reasons for these results on Hindi are not clear: maybe the semantic of Hindi language is particularly difficult to map in vectors of reals. The non-latin alphabet does not seem to be a good explaination for this loss of performance, since on the Greek language different variants of WEBELSA have an higher score than the ELSA summarizer.
A particular thing to note is that, separately, the modifications to ELSA, and so WEBELSA-RS and WEBELSA-IW, improve the performances of the summarizer, while when put together, WEBELSA-RS+IW, these leads to an impoverishment of the results. The reasons are not clear: maybe in WEBELSA-RS+IW the word embedding information is too much exploited or a totally different set of parameters should be considered. In the tables 5.2-5.9, the scores of the main algorithms are reported to have a better comprehension of the achieved results. The tables 5.10-5.17 contain the average scores of the algorithm WEBELSA-RS and WEBELSA-RS+IW, given a parameter value. The tables 5.18-5.25 contain the WEBELSA-IW average scores, given a parameter value. A larger number of parameter combinations were tested on WEBELSA-RS and WEBELSA-RS+IW because these algorithm seemed to be more promising than WEBELSA-IW. A graphical representation of the impact of the parameter *numSentences* for WEBELSA-RS and WEBELSA-RS+IW is given in the figure 4.1 Moreover, two considerations need to be done: the first is about stemming. The activation/deactivation of the stemming technique could lead to different results and should be explored. The second consideration is about unknown words: while reading the news, for each language, a set of not pre-trained words was found. For example, for the English language the set is of few tens of words, while for Greek and Hindi languages this set is composed by about 1000 words. Due to the peculiar FastText capability of creating embedding for unknown words, it would be interesting to investigate on the results with these additional embeddings.

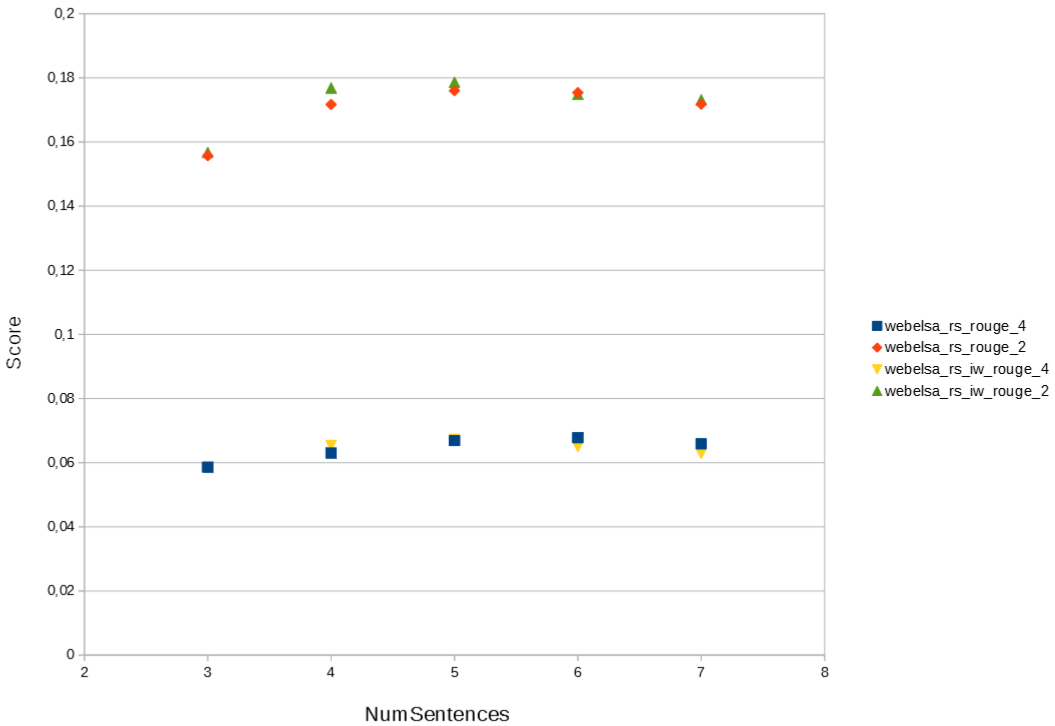

Figure 4.1: WEBELSA-RedundancyScore and WEBELSA-ItemsetWeights performances varying the number of initial read sentences.

Summarizer	Ranking Rouge-4 R measure	Ranking Rouge-2 R measure
WEBELSA-RedundancyScore-TUNED	1	1
WEBELSA-ItemsetWeights-TUNED	2	3
WEBELSA-Redundancy+Weights-TUNED	4	3
WEBELSA-RedundancyScore	6	6
WEBELSA-ItemsetWeights	5	5
WEBELSA-Redundancy+Weights	8	7
Tuned ELSA	3	2
ELSA	7	7
WBU	9	9
MWI-Sum	14	15
Maryland peer1	11	11
Maryland peer11	10	10
CIST	15	13
Maryland peer21	13	13
AMTS	16	16
ICSIsumm	12	12

Table 4.2: Rankings for Rouge-4 Recall measure and Rouge-2 Recall measure.

Table 4.3: Measures on Arabic-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 40%, minSupportPercentage=8%, numSentences=3, cosineThreshold=0.95.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 55%, minSupportPercentage=10%, numSentences=3, similarityThreshold=99.90%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 51%, minSupportPercentage=3%, numSentences=3, cosineThreshold=0.925.

Summerizer	$ROUGE-4$ R	$Confidence$ *interval*	$ROUGE-2$ R	$Confidence$ *interval*
WEBELSA-RS-TUNED	0.1494	[0.0985-0.2002]	0.2118	[0.1530-0.2706]
WEBELSA-IW-TUNED	0.1523	[0.1044-0.2002]	0.2118	[0.1560-0.2676]
WEBELSA-RS+IW-TUNED	0.1487	[0.1017-0.1958]	0.2095	[0.1547-0.2643]
WEBELSA-RS	0.1219	[0.0819-0.1619]	0.1862	[0.1399-0.2325]
WEBELSA-IW	0.1202	[0.0843-0.1561]	0.1872	[0.1465-0.2279]
WEBELSA-RS+IW	0.1163	[0.0801-0.1525]	0.1837	[0.1422-0.2252]
Tuned ELSA	0.1479	[0.1010-0.1948]	0.2085	[0.1542-0.2627]
ELSA	0.1097	[0.0738-0.1457]	0.1723	[0.1317-0.2129]
WBU	0.0997	[0.0679-0.1315]	0.1668	[0.1302-0.2035]
MWI-Sum	0.0894	[0.0537-0.1251]	0.1476	[0.1059-0.1894]
Maryland peer1	0.0715*	[0.0459-0.0971]	0.1320*	[0.1034-0.1606]
Maryland peer11	0.0673*	[0.0442-0.0904]	0.1326*	[0.1048-0.1603]
Shamoon peer5	0.0596*	[0.0378-0.0814]	0.1002*	[0.0740-0.1264]
CIST	0.0573*	[0.0326-0.0820]	0.1188*	[0.0886-0.1489]
Maryland peer21	0.0492*	[0.0290-0.0693]	0.1059*	[0.0824-0.1294]
Shamoon peer51	0.0367*	[0.0225-0.0508]	0.0856*	[0.0661-0.1051]
AMTS	0.0364*	[0.0238-0.0490]	0.0608*	[0.0436-0.0781]
Lancaster	0.0201*	[0.0092-0.0310]	0.0522*	[0.0391-0.0653]
ICSIsumm	0.0147*	[0.0045-0.0249]	0.0236*	[0.0103-0.0368]

Table 4.4: Measures on Czech-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=10%, numSentences=6, cosineThreshold=0.90.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 52%, minSupportPercentage=9%, numSentences=5, similarityThreshold=100.00%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 60%, minSupportPercentage=4%, numSentences=4, cosineThreshold=0.88.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.1152	[0.0886-0.1418]	0.2034	[0.1744-0.2323]
WEBELSA-IW-TUNED	0.1095	[0.0865-0.1324]	0.1985	[0.1728-0.2242]
WEBELSA-RS+IW-TUNED	0.1083	[0.0762-0.1403]	0.1945	[0.1576-0.2314]
WEBELSA-RS	0.1061	[0.0839-0.1284]	0.1929	[0.1680-0.2177]
WEBELSA-IW	0.1040	[0.0826-0.1254]	0.1912	[0.1679-0.2144]
WEBELSA-RS+IW	0.1075	[0.0843-0.1307]	0.1937	[0.1684-0.2189]
Tuned ELSA	0.1153	[0.0937-0.1369]	0.2046	[0.1814-0.2278]
ELSA	0.1032	[0.0834-0.1230]	0.1884	[0.1671-0.2096]
WBU	0.0934	[0.0727-0.1141]	0.1788	[0.1533-0.2042]
Maryland peer11	0.0772*	[0.0645-0.0900]	0.1662	[0.1497-0.1827]
Maryland peer21	0.0706*	[0.0560-0.0852]	0.1514*	[0.1309-0.1718]
Maryland peer1	0.0697*	[0.0532-0.0862]	0.1551*	[0.1357-0.1745]
ICSIsumm	0.0589*	[0.0449-0.0730]	0.1347*	[0.1152-0.1542]
CIST	0.0587*	[0.0432-0.0741]	0.1232*	[0.1022-0.1441]
AMTS	0.0202*	[0.0138-0.0266]	0.0513*	[0.0426-0.0601]
MWI-Sum	0.0123*	[0.0073-0.0173]	0.0341*	[0.0262-0.0419]

Table 4.5: Measures on English-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=13%, numSentences=5, cosineThreshold=0.90.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=9%, numSentences=4, similarityThreshold=99.90%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=11%, numSentences=5, cosineThreshold=0.925.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.0812	[0.0670-0.0953]	0.1969	[0.1802-0.2135]
WEBELSA-IW-TUNED	0.0827	[0.0684-0.0970]	0.2051	[0.1877-0.2225]
WEBELSA-RS+IW-TUNED	0.0788	[0.0654-0.0922]	0.1984	[0.1827-0.2141]
WEBELSA-RS	0.0737	[0.0586-0.0888]	0.1897	[0.1709-0.2084]
WEBELSA-IW	0.0795	[0.0678-0.0912]	0.2009	[0.1854-0.2164]
WEBELSA-RS+IW	0.0722	[0.0612-0.0832]	0.1911	[0.1763-0.2059]
Tuned ELSA	0.0746	[0.0635-0.0857]	0.1862	[0.1708-0.2016]
ICSIsumm	0.0713	[0.0591-0.0836]	0.1938	[0.1766-0.2111]
ELSA	0.0692	[0.0533-0.0850]	0.1829	[0.1639-0.2018]
WBU	0.0640	[0.0533-0.0747]	0.1711	[0.1573-0.1848]
Maryland peer1	0.0502	[0.0398-0.0606]	0.1614	[0.1438-0.1791]
Maryland peer11	0.0492	[0.0391-0.0593]	0.1610	[0.1447-0.1774]
MWI-Sum	0.0462	[0.0348-0.0576]	0.1389*	[0.1202-0.1577]
CIST	0.0447*	[0.0356-0.0538]	0.1467*	[0.1329-0.1605]
Maryland peer21	0.0415*	[0.0319-0.0512]	0.1418*	[0.1252-0.1584]
Coverage	0.0335*	[0.0208-0.0462]	0.0931*	[0.0686-0.1176]
Shamoon peer5	0.0315*	[0.0201-0.0428]	0.1105*	[0.0921-0.1289]
ClusterCMRW	0.0259*	[0.0138-0.0381]	0.0794*	[0.0578-0.1009]
TextRank	0.0248*	[0.0141-0.0355]	0.0716*	[0.0497-0.0935]
Centroid	0.0241*	[0.0141-0.0342]	0.0704*	[0.0503-0.0906]
Submodular	0.0237*	[0.0113-0.0362]	0.0813*	[0.0587-0.1040]
AMTS	0.0224*	[0.0166-0.0283]	0.0567*	[0.0454-0.0680]
Shamoon peer51	0.0224*	[0.0178-0.0270]	0.0965*	[0.0870-0.1060]
LexPageRank	0.0211*	[0.0130-0.0293]	0.0690*	[0.0525-0.0855]
ILP	0.0204*	[0.0121-0.0287]	0.0763*	[0.0559-0.0967]
Lancaster	0.0169*	[0.0121-0.0218]	0.0828*	[0.0745-0.0912]
Lead	0.0128*	[0.0068-0.0187]	0.0565*	[0.0429-0.0701]

Table 4.6: Measures on French-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 43%, minSupportPercentage=11%, numSentences=3, cosineThreshold=0.95.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 52%, minSupportPercentage=14%, numSentences=3, similarityThreshold=100.00%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 52%, minSupportPercentage=6%, numSentences=3, cosineThreshold=0.925.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.1243	[0.089-0.1647]	0.2356	[0.2013-0.2700]
WEBELSA-IW-TUNED	0.1241	[0.0877-0.1656]	0.2387	[0.2040-0.2734]
WEBELSA-RS+IW-TUNED	0.1237	[0.0845-0.1629]	0.2402	[0.2033-0.2771]
WEBELSA-RS	0.1110	[0.0809-0.1412]	0.2197	[0.1929-0.2464]
WEBELSA-IW	0.1153	[0.0863-0.1443]	0.2253	[0.1985-0.2521]
WEBELSA-RS+IW	0.1126	[0.0842-0.1409]	0.2293	[0.2007-0.2578]
Tuned ELSA	0.1168	[0.0810-0.1527]	0.2309	[0.1982-0.2636]
ELSA	0.1063	[0.0797-0.1328]	0.2167	[0.1941-0.2394]
ICSIsumm	0.1012	[0.0698-0.1327]	0.2243	[0.1902-0.2584]
WBU	0.1007	[0.0606-0.1408]	0.2134	[0.1746-0.2522]
Maryland peer11	0.0816	[0.0569-0.1063]	0.2013	[0.1683-0.2343]
Maryland peer1	0.0806	[0.0566-0.1045]	0.2008	[0.1724-0.2292]
MWI-Sum	0.0768*	[0.0596-0.0939]	0.1858*	[0.1633-0.2083]
Maryland peer21	0.0625*	[0.0418-0.0832]	0.1772*	[0.1517-0.2027]
Coverage	0.0585*	[0.0398-0.0772]	0.1412*	[0.1210-0.1613]
Submodular	0.0570*	[0.0320-0.0819]	0.1383*	[0.1071-0.1695]
ClusterCMRW	0.0564*	[0.0322-0.0805]	0.1418*	[0.1147-0.1689]
CIST	0.0560*	[0.0371-0.0749]	0.1660*	[0.1415-0.1904]
ILP	0.0480*	[0.0332-0.0629]	0.1390*	[0.1223-0.1557]
AMTS	0.0463*	[0.0329-0.0598]	0.1006*	[0.0822-0.1191]
TextRank	0.0403*	[0.0248-0.0559]	0.1061*	[0.0882-0.1240]
LexPageRank	0.0373*	[0.0161-0.0585]	0.1086*	[0.0853-0.1319]
Centroid	0.0311*	[0.0209-0.0414]	0.0920*	[0.0730-0.1109]
Lead	0.0265*	[0.0134-0.0396]	0.0868*	[0.0722-0.1014]

Table 4.7: Measures on Greek-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 45%, minSupportPercentage=2%, numSentences=6, cosineThreshold=0.90.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 80%, minSupportPercentage=6%, numSentences=4, similarityThreshold=99.90%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 80%, minSupportPercentage=6%, numSentences=4, cosineThreshold=0.895.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.0467	[0.0355-0.0578]	0.1412	[0.1254-0.1570]
WEBELSA-IW-TUNED	0.0397	[0.0295-0.0498]	0.1293	[0.1142-0.1445]
WEBELSA-RS+IW-TUNED	0.0392	[0.0291-0.0493]	0.1306	[0.1158-0.1454]
WEBELSA-RS	0.0375	[0.0294-0.0456]	0.1297	[0.1170-0.1424]
WEBELSA-IW	0.0380	[0.0293-0.0466]	0.1246	[0.1125-0.1367]
WEBELSA-RS+IW	0.0352	[0.0268-0.0436]	0.1201	[0.1088-0.1315]
Tuned ELSA	0.0390	[0.0297-0.0484]	0.1294	[0.1169-0.1418]
ELSA	0.0338	[0.0251-0.0426]	0.1243	[0.1118-0.1368]
Maryland peer11	0.0329	[0.0221-0.0438]	0.1237	[0.1067-0.1406]
Maryland peer1	0.0313	[0.0221-0.0405]	0.1200	[0.1063-0.1336]
Maryland peer21	0.0312	[0.0215-0.0409]	0.1117	[0.0960-0.1274]
MWI-Sum	0.0285	[0.0204-0.0366]	0.1074	[0.0948-0.1199]
WBU	0.0260	[0.0188-0.0332]	0.1104	[0.1001-0.1206]
CIST	0.0195*	[0.0134-0.0256]	0.1002	[0.0865-0.1139]
ICSIsumm	0.0142*	[0.0087-0.0198]	0.0567*	[0.0409-0.0726]
AMTS	0.0142*	[0.0081-0.0202]	0.0512*	[0.0384-0.0639]

Table 4.8: Measures on Hindi-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 110%, minSupportPercentage=2%, numSentences=3, cosineThreshold=0.90.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=5%, numSentences=6, similarityThreshold=99.90%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=5%, numSentences=6, cosineThreshold=0.88.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.0983	[0.0696-0.1269]	0.2975	[0.2645-0.3304]
WEBELSA-IW-TUNED	0.0891	[0.0625-0.1157]	0.3004	[0.2695-0.3313]
WEBELSA-RS+IW-TUNED	0.0966	[0.0692-0.1241]	0.3099	[0.2785-0.3412]
WEBELSA-RS	0.0781	[0.0496-0.1066]	0.2735	[0.2396-0.3073]
WEBELSA-IW	0.0758	[0.0477-0.1039]	0.2779	[0.2424-0.3134]
WEBELSA-RS+IW	0.0754	[0.0512-0.0996]	0.2751	[0.2435-0.3066]
Tuned ELSA	0.1144	[0.0907-0.1381]	0.3680	[0.3403-0.3956]
ELSA	0.0961	[0.0742-0.1181]	0.3454	[0.3176-0.3732]
WBU	0.0934	[0.0727-0.1141]	0.3192	[0.2882-0.3503]
Maryland peer11	0.0874	[0.0712-0.1036]	0.3439	[0.3206-0.3672]
CIST	0.0809	[0.0641-0.0976]	0.3456	[0.3181-0.3731]
Maryland peer21	0.0800	[0.0667-0.0932]	0.3285	[0.3087-0.3482]
Maryland peer1	0.0795	[0.0687-0.0903]	0.3348	[0.3128-0.3567]
MWI-Sum	0.0679*	[0.0431-0.0926]	0.2425*	[0.1901-0.2949]
AMTS	0.0464*	[0.0270-0.0658]	0.1603*	[0.1096-0.2111]
ICSIsumm	0.0091*	[-0.0008-0.0191]	0.0482*	[0.0039-0.0925]

Table 4.9: Measures on Romanian-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 77%, minSupportPercentage=11%, numSentences=6, cosineThreshold=0.925. WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 50%, minSupportPercentage=7%, numSentences=5, similarityThreshold=99.90%. WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 110%, minSupportPercentage=6%, numSentences=4, cosineThreshold=0.925.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.0998	[0.0620-0.1376]	0.1961	[0.1557-0.2365]
WEBELSA-IW-TUNED	0.0871	[0.0538-0.1203]	0.1766	[0.1433-0.2099]
WEBELSA-RS+IW-TUNED	0.0844	[0.0496-0.1192]	0.1784	[0.1418-0.2150]
WEBELSA-RS	0.0692	[0.0408-0.0975]	0.1564	[0.1287-0.1841]
WEBELSA-IW	0.0871	[0.0538-0.1203]	0.1766	[0.1433-0.2099]
WEBELSA-RS+IW	0.0849	[0.0500-0.1197]	0.1751	[0.1374-0.2128]
Tuned ELSA	0.0960	[0.0623-0.1298]	0.1910	[0.1569-0.2250]
ELSA	0.0760	[0.0478-0.1042]	0.1637	[0.1345-0.1928]
WBU	0.0681	[0.0406-0.0955]	0.1655	[0.1329-0.1982]
Maryland peer21	0.0496	[0.0327-0.0664]	0.1381	[0.1182-0.1580]
Maryland peer1	0.0493*	[0.0308-0.0677]	0.1489	[0.1240-0.1738]
ICSIsumm	0.0469*	[0.0303-0.0636]	0.1504	[0.1307-0.1701]
Maryland peer11	0.0437*	[0.0317-0.0558]	0.1392	[0.1222-0.1563]
CIST	0.0385*	[0.0233-0.0538]	0.1202*	[0.0965-0.1439]
AMTS	0.0332*	[0.0177-0.0487]	0.0714*	[0.0519-0.0909]
MWI-Sum	0.0152*	[0.0061-0.0242]	0.0461*	[0.0299-0.0624]

Table 4.10: Measures on Spanish-written Multiling 2013 document collection. WEBELSA-RedundancyScore-TUNED configuration: overlapTheshold= 60%, minSupportPercentage=10%, numSentences=4, cosineThreshold=0.95.
WEBELSA-ItemsetWeights-TUNED configuration: overlapTheshold= 62%, minSupportPercentage=13%, numSentences=3, similarityThreshold=99.90%.
WEBELSA-Redundancy+Weights-TUNED configuration: overlapTheshold= 70%, minSupportPercentage=8%, numSentences=3, cosineThreshold=0.925.

Summerizer	$ROUGE-4$ R	*Confidence interval*	$ROUGE-2$ R	*Confidence interval*
WEBELSA-RS-TUNED	0.1386	[0.1199-0.1573]	0.2752	[0.2569-0.2935]
WEBELSA-IW-TUNED	0.1434	[0.1135-0.1673]	0.2738	[0.2474-0.3002]
WEBELSA-RS+IW-TUNED	0.1391	[0.1150-0.1631]	0.2719	[0.2464-0.2975]
WEBELSA-RS	0.1144	[0.0970-0.1318]	0.2468	[0.2272-0.2664]
WEBELSA-IW	0.0983	[0.0831-0.1136]	0.2245	[0.2052-0.2439]
WEBELSA-RS+IW	0.1017	[0.0893-0.1140]	0.2284	[0.2113-0.2454]
Tuned ELSA	0.1408	[0.1223-0.1592]	0.2773	[0.2594-0.2951]
ELSA	0.1145	[0.0969-0.1320]	0.2458	[0.2255-0.2661]
MWI-Sum	0.1133	[0.0952-0.1314]	0.2377	[0.2147-0.2607]
ICSIsumm	0.1116	[0.0939-0.1293]	0.2565	[0.2362-0.2769]
WBU	0.1039	[0.0885-0.1193]	0.2274	[0.2095-0.2453]
Coverage	0.1008	[0.0808-0.1207]	0.1994*	[0.1754-0.2233]
Maryland peer11	0.0877*	[0.0747-0.1008]	0.2180	[0.2008-0.2352]
ILP	0.0716*	[0.0522-0.0910]	0.1723*	[0.1450-0.1995]
ClusterCMRW	0.0705*	[0.0538-0.0872]	0.1702*	[0.1514-0.1891]
Maryland peer1	0.0670*	[0.0572-0.0767]	0.1975*	[0.1807-0.2143]
CIST	0.0669*	[0.0501-0.0838]	0.1801*	[0.1578-0.2024]
Submodular	0.0625*	[0.0515-0.0736]	0.1647*	[0.1490-0.1803]
Maryland peer21	0.0596*	[0.0447-0.0745]	0.1754*	[0.1543-0.1965]
AMTS	0.0534*	[0.0429-0.0640]	0.1011*	[0.0867-0.1155]
Centroid	0.0488*	[0.0335-0.0641]	0.1308*	[0.1093-0.1522]
TextRank	0.0474*	[0.0354-0.0593]	0.1357*	[0.1164-0.1551]
Lead	0.0469*	[0.0362-0.0575]	0.1198*	[0.1045-0.1352]
LexPageRank	0.0428*	[0.0315-0.0541]	0.1253*	[0.1100-0.1406]

Table 4.11: Average measures for Arabic document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$		$WEBELSA-RS+IW$	
		ROUGE-4	ROUGE-2	ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.1068	0.1706	0.1109	0.1765
SimilarityMeasure=cosine	0.90	0.1239	0.1870	0.1227	0.1853
SimilarityMeasure=cosine	0.925	0.1272	0.1893	0.1224	0.1863
SimilarityMeasure=cosine	0.95	0.1251	0.1866	0.1213	0.1839
overlapThreshold	50	0.1238	0.1863	0.1220	0.1863
overlapThreshold	60	0.1193	0.1814	0.1213	0.1849
overlapThreshold	70	0.1209	0.1839	0.1208	0.1850
overlapThreshold	80	0.1201	0.1827	0.1185	0.1828
overlapThreshold	90	0.1204	0.1827	0.1175	0.1817
overlapThreshold	110	0.1204	0.1836	0.1165	0.1802
numSentences	3	0.1340	0.1945	0.1319	0.1930
numSentences	4	0.1250	0.1875	0.1245	0.1883
numSentences	5	0.1158	0.1800	0.1130	0.1793
numSentences	6	0.1113	0.1745	0.1111	0.1766
numSentences	7	0.1074	0.1705	0.1067	0.1711
minSupportPercentage	2	0.1239	0.1862	0.1227	0.1864
minSupportPercentage	3	0.1230	0.1854	0.1217	0.1855
minSupportPercentage	4	0.1190	0.1818	0.1181	0.1822
minSupportPercentage	5	0.1192	0.1819	0.1177	0.1819
minSupportPercentage	6	0.1204	0.1832	0.1183	0.1826

Table 4.12: Average measures for Czech document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$ ROUGE-4	ROUGE-2	$WEBELSA-RS+IW$ ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.0841	0.1598	0.0806	0.1578
SimilarityMeasure=cosine	0.90	0.0835	0.1593	0.0806	0.1577
SimilarityMeasure=cosine	0.925	0.0830	0.1585	0.0794	0.1567
SimilarityMeasure=cosine	0.95	0.0828	0.1582	0.0787	0.1560
overlapThreshold	50	0.0863	0.1625	0.0838	0.1608
overlapThreshold	60	0.0850	0.1626	0.0825	0.1608
overlapThreshold	70	0.0841	0.1601	0.0787	0.1567
overlapThreshold	80	0.0827	0.1576	0.0780	0.1550
overlapThreshold	90	0.0812	0.1558	0.0783	0.1550
overlapThreshold	110	0.0806	0.1550	0.0777	0.1543
numSentences	3	0.0160	0.0575	0.0159	0.0579
numSentences	4	0.1000	0.1840	0.1011	0.1857
numSentences	5	0.1035	0.1871	0.0994	0.1848
numSentences	6	0.1013	0.1870	0.0918	0.1796
numSentences	7	0.0958	0.1791	0.0911	0.1773
minSupportPercentage	2	0.0819	0.1569	0.0793	0.1563
minSupportPercentage	3	0.0819	0.1569	0.0793	0.1563
minSupportPercentage	4	0.0829	0.1585	0.0801	0.1572
minSupportPercentage	5	0.0847	0.1609	0.0801	0.1576
minSupportPercentage	6	0.0852	0.1615	0.0803	0.1579

Table 4.13: Average measures for English document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$ ROUGE-4	ROUGE-2	$WEBELSA-RS+IW$ ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.0628	0.1646	0.0612	0.1651
SimilarityMeasure=cosine	0.90	0.0690	0.1766	0.0655	0.1748
SimilarityMeasure=cosine	0.925	0.0636	0.1705	0.0642	0.1740
SimilarityMeasure=cosine	0.95	0.0623	0.1688	0.0642	0.1740
overlapThreshold	50	0.0656	0.1731	0.0677	0.1776
overlapThreshold	60	0.0668	0.1735	0.0686	0.1785
overlapThreshold	70	0.0639	0.1696	0.0643	0.1728
overlapThreshold	80	0.0636	0.1688	0.0624	0.1698
overlapThreshold	90	0.0640	0.1690	0.0603	0.1672
overlapThreshold	110	0.0628	0.1667	0.0595	0.1660
numSentences	3	0.0586	0.1557	0.0584	0.1567
numSentences	4	0.0630	0.1717	0.0654	0.1768
numSentences	5	0.0669	0.1760	0.0673	0.1786
numSentences	6	0.0678	0.1754	0.0650	0.1748
numSentences	7	0.0659	0.1718	0.0629	0.1731
minSupportPercentage	2	0.0644	0.1698	0.0645	0.1726
minSupportPercentage	3	0.0644	0.1698	0.0644	0.1727
minSupportPercentage	4	0.0645	0.1700	0.0638	0.1718
minSupportPercentage	5	0.0645	0.1706	0.0635	0.1718
minSupportPercentage	6	0.0645	0.1703	0.0627	0.1711

Table 4.14: Average measures for French document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA - RS$		$WEBELSA - RS + IW$	
		ROUGE-4	ROUGE-2	ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.0743	0.1869	0.0764	0.1912
SimilarityMeasure=cosine	0.90	0.0857	0.2013	0.0937	0.2083
SimilarityMeasure=cosine	0.925	0.0980	0.2065	0.1035	0.2186
SimilarityMeasure=cosine	0.95	0.1012	0.2084	0.1055	0.2191
overlapThreshold	50	0.0915	0.2004	0.0999	0.2124
overlapThreshold	60	0.0899	0.2001	0.0954	0.2109
overlapThreshold	70	0.0916	0.2016	0.0948	0.2104
overlapThreshold	80	0.0908	0.2006	0.0947	0.2097
overlapThreshold	90	0.0914	0.2017	0.0919	0.2068
overlapThreshold	110	0.0898	0.1994	0.0920	0.2077
numSentences	3	0.1006	0.2075	0.0935	0.2045
numSentences	4	0.0934	0.2063	0.0904	0.2079
numSentences	5	0.0875	0.1998	0.0987	0.2158
numSentences	6	0.0897	0.2000	0.0958	0.2105
numSentences	7	0.0841	0.1900	0.0956	0.2095
minSupportPercentage	2	0.0917	0.2024	0.0942	0.2091
minSupportPercentage	3	0.0913	0.2019	0.0941	0.2092
minSupportPercentage	4	0.0897	0.1993	0.0944	0.2092
minSupportPercentage	5	0.0897	0.1988	0.0952	0.2101
minSupportPercentage	6	0.0918	0.2009	0.0960	0.2106

Table 4.15: Average measures for Greek document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$ ROUGE-4	ROUGE-2	$WEBELSA-RS+IW$ ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.0380	0.1291	0.0308	0.1088
SimilarityMeasure=cosine	0.90	0.0380	0.1285	0.0322	0.1114
SimilarityMeasure=cosine	0.925	0.0365	0.1252	0.0321	0.1110
SimilarityMeasure=cosine	0.95	0.0349	0.1224	0.0313	0.1095
overlapThreshold	50	0.0373	0.1266	0.0302	0.1053
overlapThreshold	60	0.0373	0.1262	0.0317	0.1094
overlapThreshold	70	0.0366	0.1262	0.0317	0.1102
overlapThreshold	80	0.0369	0.1266	0.0320	0.1120
overlapThreshold	90	0.0364	0.1262	0.0319	0.1121
overlapThreshold	110	0.0366	0.1261	0.0319	0.1121
numSentences	3	0.0334	0.1206	0.0223	0.0860
numSentences	4	0.0391	0.1301	0.0359	0.1231
numSentences	5	0.0372	0.1275	0.0354	0.1188
numSentences	6	0.0383	0.1282	0.0348	0.1163
numSentences	7	0.0360	0.1251	0.0294	0.1067
minSupportPercentage	2	0.0370	0.1259	0.0322	0.1119
minSupportPercentage	3	0.0369	0.1259	0.0321	0.1119
minSupportPercentage	4	0.0371	0.1263	0.0317	0.1109
minSupportPercentage	5	0.0368	0.1268	0.0315	0.1096
minSupportPercentage	6	0.0364	0.1266	0.0303	0.1065

Table 4.16: Average measures for Hindi document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$ ROUGE-4	ROUGE-2	$WEBELSA-RS+IW$ ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.0824	0.2795	0.0857	0.2913
SimilarityMeasure=cosine	0.90	0.0826	0.2805	0.0830	0.2869
SimilarityMeasure=cosine	0.925	0.0816	0.2767	0.0809	0.2834
SimilarityMeasure=cosine	0.95	0.0811	0.2767	0.0805	0.2828
overlapThreshold	50	0.0797	0.2761	0.0816	0.2877
overlapThreshold	60	0.0816	0.278	0.0843	0.2883
overlapThreshold	70	0.0820	0.2795	0.0825	0.285
overlapThreshold	80	0.0801	0.2766	0.0812	0.2849
overlapThreshold	90	0.0821	0.2772	0.0810	0.2837
overlapThreshold	110	0.0861	0.2827	0.0844	0.2871
numSentences	3	0.0900	0.289	0.0928	0.2985
numSentences	4	0.0820	0.2781	0.0823	0.2843
numSentences	5	0.0812	0.2779	0.0801	0.2844
numSentences	6	0.0789	0.275	0.0794	0.2829
numSentences	7	0.0775	0.2717	0.0780	0.2803
minSupportPercentage	2	0.0824	0.2793	0.0815	0.2842
minSupportPercentage	3	0.0817	0.2782	0.0814	0.2845
minSupportPercentage	4	0.0816	0.2785	0.0831	0.2881
minSupportPercentage	5	0.0817	0.2781	0.0831	0.2874
minSupportPercentage	6	0.0822	0.2777	0.0835	0.2864

Table 4.17: Average measures for Romanian document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$		$WEBELSA-RS+IW$	
		ROUGE-4	ROUGE-2	ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.0649	0.1478	0.0430	0.1243
SimilarityMeasure=cosine	0.90	0.0729	0.1601	0.0499	0.1339
SimilarityMeasure=cosine	0.925	0.0803	0.1715	0.0659	0.1500
SimilarityMeasure=cosine	0.95	0.0757	0.1651	0.0676	0.1522
overlapThreshold	50	0.0726	0.1577	0.0640	0.1481
overlapThreshold	60	0.0718	0.1608	0.0473	0.1281
overlapThreshold	70	0.0737	0.1633	0.0464	0.1264
overlapThreshold	80	0.0749	0.1647	0.0491	0.1317
overlapThreshold	90	0.0738	0.1633	0.0540	0.1388
overlapThreshold	110	0.0749	0.1639	0.0540	0.1388
numSentences	3	0.0719	0.1531	0.0739	0.1555
numSentences	4	0.0723	0.1607	0.0597	0.1493
numSentences	5	0.0748	0.1637	0.0667	0.1547
numSentences	6	0.0793	0.1693	0.0555	0.1389
numSentences	7	0.0684	0.1546	0.0438	0.1225
minSupportPercentage	2	0.0752	0.1602	0.0677	0.1529
minSupportPercentage	3	0.0745	0.1596	0.0643	0.1489
minSupportPercentage	4	0.0733	0.1603	0.0558	0.1389
minSupportPercentage	5	0.0734	0.1618	0.0538	0.1368
minSupportPercentage	6	0.0729	0.1617	0.0529	0.1360

Table 4.18: Average measures for Spanish document collections for WEBELSA-RedundancyScore and WEBELSA-Redundancy+Weights.
The values of overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$WEBELSA-RS$		$WEBELSA-RS+IW$	
		ROUGE-4	ROUGE-2	ROUGE-4	ROUGE-2
SimilarityMeasure=cosine	0.88	0.1002	0.2212	0.0985	0.2190
SimilarityMeasure=cosine	0.90	0.1094	0.2362	0.1063	0.2315
SimilarityMeasure=cosine	0.925	0.1177	0.2457	0.1116	0.2391
SimilarityMeasure=cosine	0.95	0.1176	0.2447	0.1104	0.2356
overlapThreshold	50	0.1028	0.2249	0.1010	0.2218
overlapThreshold	60	0.1128	0.2378	0.1074	0.2312
overlapThreshold	70	0.1125	0.2387	0.1086	0.2333
overlapThreshold	80	0.1123	0.2386	0.1072	0.2331
overlapThreshold	90	0.1134	0.2406	0.1076	0.2342
overlapThreshold	110	0.1136	0.2409	0.1086	0.2350
numSentences	3	0.1204	0.2437	0.1195	0.2429
numSentences	4	0.1247	0.2565	0.1169	0.2478
numSentences	5	0.1181	0.2487	0.1076	0.2359
numSentences	6	0.1018	0.2254	0.0998	0.2230
numSentences	7	0.0908	0.2098	0.0892	0.2066
minSupportPercentage	2	0.1102	0.2355	0.1074	0.2328
minSupportPercentage	3	0.1102	0.2356	0.1071	0.2322
minSupportPercentage	4	0.1113	0.2371	0.1068	0.2315
minSupportPercentage	5	0.1119	0.2377	0.1063	0.2306
minSupportPercentage	6	0.1126	0.2386	0.1059	0.2299

Table 4.19: Average measures for Arabic document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	*ROUGE − 4* R	*ROUGE − 2* R
SimilarityMeasure=similarityThreshold	95	0.0577	0.0954
SimilarityMeasure=similarityThreshold	99.9	0.1194	0.1841
SimilarityMeasure=similarityThreshold	100	0.1206	0.1845
overlapThreshold	50	0.1223	0.1886
overlapThreshold	60	0.1234	0.1875
overlapThreshold	70	0.1187	0.1826
overlapThreshold	80	0.1157	0.1789
numSentences	4	0.1276	0.1912
numSentences	5	0.1149	0.1800
numSentences	6	0.1144	0.1783
minSupportPercentage	1	0.1190	0.1832
minSupportPercentage	3	0.1190	0.1832
minSupportPercentage	5	0.1184	0.1824
minSupportPercentage	7	0.1228	0.1875

Table 4.20: Average measures for Czech document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	*ROUGE − 4* R	*ROUGE − 2* R
SimilarityMeasure=similarityThreshold	95	0.0643	0.1267
SimilarityMeasure=similarityThreshold	99.9	0.0953	0.1815
SimilarityMeasure=similarityThreshold	100	0.0979	0.1846
overlapThreshold	50	0.1011	0.1870
overlapThreshold	60	0.0989	0.1864
overlapThreshold	70	0.0936	0.1806
overlapThreshold	80	0.0928	0.1781
numSentences	4	0.0998	0.1849
numSentences	5	0.0988	0.1849
numSentences	6	0.0912	0.1793
minSupportPercentage	1	0.0968	0.1832
minSupportPercentage	3	0.0968	0.1832
minSupportPercentage	5	0.0963	0.1829
minSupportPercentage	7	0.0965	0.1828

Table 4.21: Average measures for English document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	$ROUGE-4$ R	$ROUGE-2$ R
SimilarityMeasure=similarityThreshold	95	0.0253	0.0816
SimilarityMeasure=similarityThreshold	99.9	0.0685	0.1805
SimilarityMeasure=similarityThreshold	100	0.0668	0.1786
overlapThreshold	50	0.0728	0.1886
overlapThreshold	60	0.0716	0.1849
overlapThreshold	70	0.0640	0.1743
overlapThreshold	80	0.0621	0.1705
numSentences	4	0.0658	0.1779
numSentences	5	0.0698	0.1826
numSentences	6	0.0673	0.1781
minSupportPercentage	1	0.0680	0.1800
minSupportPercentage	3	0.0680	0.1800
minSupportPercentage	5	0.0674	0.1791
minSupportPercentage	7	0.0672	0.1791

Table 4.22: Average measures for French document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

$Parameter$	$Value$	$ROUGE-4$ R	$ROUGE-2$ R
SimilarityMeasure=similarityThreshold	95	0.0487	0.1195
SimilarityMeasure=similarityThreshold	99.9	0.1043	0.2176
SimilarityMeasure=similarityThreshold	100	0.1050	0.2179
overlapThreshold	50	0.1099	0.2228
overlapThreshold	60	0.1031	0.2181
overlapThreshold	70	0.1036	0.2163
overlapThreshold	80	0.1018	0.2139
numSentences	4	0.1049	0.2184
numSentences	5	0.1081	0.2220
numSentences	6	0.1008	0.2129
minSupportPercentage	1	0.1056	0.2186
minSupportPercentage	3	0.1056	0.2186
minSupportPercentage	5	0.1024	0.2169
minSupportPercentage	7	0.1048	0.2170

Table 4.23: Average measures for Greek document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	*ROUGE − 4* R	*ROUGE − 2* R
SimilarityMeasure=similarityThreshold	95	0.0258	0.0969
SimilarityMeasure=similarityThreshold	99.9	0.0342	0.1167
SimilarityMeasure=similarityThreshold	100	0.0341	0.1160
overlapThreshold	50	0.0331	0.1145
overlapThreshold	60	0.0343	0.1161
overlapThreshold	70	0.0344	0.1162
overlapThreshold	80	0.0349	0.1188
numSentences	4	0.0344	0.1181
numSentences	5	0.0350	0.1177
numSentences	6	0.0331	0.1133
minSupportPercentage	1	0.0350	0.1189
minSupportPercentage	3	0.0350	0.1189
minSupportPercentage	5	0.0341	0.1161
minSupportPercentage	7	0.0325	0.1115

Table 4.24: Average measures for Hindi document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	*ROUGE − 4* R	*ROUGE − 2* R
SimilarityMeasure=similarityThreshold	95	0.0568	0.2524
SimilarityMeasure=similarityThreshold	99.9	0.0790	0.2815
SimilarityMeasure=similarityThreshold	100	0.0781	0.2804
overlapThreshold	50	0.0752	0.2779
overlapThreshold	60	0.0825	0.2848
overlapThreshold	70	0.0789	0.2803
overlapThreshold	80	0.0777	0.2809
numSentences	4	0.0793	0.2806
numSentences	5	0.0782	0.2816
numSentences	6	0.0782	0.2807
minSupportPercentage	1	0.0765	0.2785
minSupportPercentage	3	0.0765	0.2785
minSupportPercentage	5	0.0796	0.2840
minSupportPercentage	7	0.0817	0.2830

Table 4.25: Average measures for Romanian document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	*ROUGE* − 4 R	*ROUGE* − 2 R
SimilarityMeasure=similarityThreshold	95	0.0235	0.0662
SimilarityMeasure=similarityThreshold	99.9	0.0707	0.1579
SimilarityMeasure=similarityThreshold	100	0.0735	0.1599
overlapThreshold	50	0.0740	0.1620
overlapThreshold	60	0.0670	0.1510
overlapThreshold	70	0.0678	0.1509
overlapThreshold	80	0.0708	0.1570
numSentences	4	0.0691	0.1596
numSentences	5	0.0793	0.1671
numSentences	6	0.0669	0.1502
minSupportPercentage	1	0.0766	0.1656
minSupportPercentage	3	0.0766	0.1656
minSupportPercentage	5	0.0681	0.1529
minSupportPercentage	7	0.0766	0.1656

Table 4.26: Average measures for Spanish document collections for WEBELSA-ItemsetWeights.
The values of similarityThreshold, overlapThreshold and minSupportPercentage are percentages.

Parameter	*Value*	*ROUGE − 4* R	*ROUGE − 2* R
SimilarityMeasure=similarityThreshold	95	0.0410	0.0994
SimilarityMeasure=similarityThreshold	99.9	0.1100	0.2375
SimilarityMeasure=similarityThreshold	100	0.1104	0.2372
overlapThreshold	50	0.1022	0.2305
overlapThreshold	60	0.1122	0.2395
overlapThreshold	70	0.1150	0.2419
overlapThreshold	80	0.1123	0.2383
numSentences	4	0.1191	0.2495
numSentences	5	0.1093	0.2366
numSentences	6	0.0999	0.2228
minSupportPercentage	1	0.1097	0.2381
minSupportPercentage	3	0.1095	0.2376
minSupportPercentage	5	0.1101	0.2374
minSupportPercentage	7	0.1112	0.2365

Chapter 5

Conclusions and future work

This work is a preliminary attempt to apply vector representations of word to drive the document summarization process. The results achieved on benchmark multilingual data show better performance than state-of-the-art approaches. However, an open issue is how to properly set the input parameters. Despite the standard configurations already achieved promising performance, still parameter tuning allows further improving the algorithm performance. About this thesis, more parameter combinations could be investigated to have a better idea of the best-mean values to chose. In fact, from the tables it is possible to see that some tuned variants have parameters that are very far from the ones described and recommended in this thesis. Moreover, it would be useful to understand how performances range when embeddings for unknown words are computed, because for some languages there are more than 1000 unknown words. For this thesis, just the cosine similarity among word vectors was exploited to measure how two words are related: it would be useful to use different types of measure. Word embedding is a new field that is very interesting, for this reason new models often are released: these models could be able to store and capture more information than FastText model and so be exploited. Finally, sentence embedding seems to be one of the most promising vehicle to improve performances. In this regard, many works could be mentioned: one of these is the model Sent2vec of Pagliardini et al. [46], but unfortunately sentence embedding are techniques that do not seem to be ready to be used, from a computational point of view.

Bibliography

[1] Luca Cagliero, Paolo Garza, and Elena Baralis, *ELSA: A Multilingual Document Summarization Algorithm Based on Frequent Itemsets and Latent Semantic Analysis*, Politecnico di Torino.

[2] Elena Baralis, Luca Cagliero, Naeem Mahoto, Alessandro Fiori. *Discovering correlations among multiple terms for graph-based summarization* Information Sciences ELSEVIER DOI: 10.1016/j.ins.2013.06.046).

[3] Elena Baralis, Luca Cagliero, Alessandro Fiori, and Paolo Garza. *2015. MWI-Sum: A Multilingual Summarizer Based on Frequent Weighted Itemsets. ACM Trans. Inf. Syst. 34, 1, Article 5 (September 2015), 35 pages. DOI: https://doi.org/10.1145/2809786).*

[4] Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig *Proceedings of NAACL-HLT 2013, pages 746–751,Atlanta, Georgia, 9–14 June 2013.c2013 Association for Computational Linguistics Linguistic Regularities in Continuous Space Word Representations Microsoft Research Redmond, WA 98052.*

[5] Pennington, Richard Socher, Christopher D. Manning *GloVe: Global Vectors for Word RepresentationJeffrey Computer Science Department, Stanford University, Stanford, CA 94305.*

[6] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova *BERT: Pre-training of Deep Bidirectional Transformers forLanguage Understanding Google AI Language.*

[7] Piotr Bojanowski and Edouard Grave and Armand Joulin and Tomas Mikolov, *Enriching Word Vectors with Subword Information*, Facebook AI Research,19 June 2017.

[8] Lucas de Haas, *Extractive Summarization using sentence embeddings* Department of Information and Computing Sciences, Faculty of Science, Utrecht University, November 29, 2017.

[9] Lin, H. and Bilmes, J. (2010). *Multi-document summarization via budgeted maximization of submodular functions*, In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 912–920. Association for Computational Linguistics.

[10] Lin, H. and Bilmes, J. (2011). A class of submodular functions for document summarization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, volume 1, pages 510–520. Association for Computational Linguistics.

[11] Kågebäck, M., Mogren, O., Tahmasebi, N., and Dubhashi, D. (2014). *Extractive summarization using continuous vector space models, In Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL, pages 31–39. Citeseer.*

[12] Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., and Manning, C. D. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection, In NIPS, volume 24, pages 801–809.

[13] Kobayashi, H., Yatsuka, M., and Taichi, N. (2015). *Summarization Based on Embedding Distributions*, EMNLP, pages 1984–1989.

[14] Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. *Distributed representations of words and phrases and their compositionality, In Adv. NIPS.*

[15] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, Tomas Mikolov Learning Word Vectors for 157 Languages, Facebook AI Research, EPFL, 28 Mar 2018.

[16] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. 2007. Frequent pattern mining: Current status and future directions. Data Min. Knowl. Discov. 15, 1 (2007), 55–86.

[17] Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In Proceedings of the 18th Conference on Computational Linguistics - Volume 1 (COLING'00). Association for Computational Linguistics, Stroudsburg, PA, 495–501. DOI:http://dx.doi.org/10.3115/990820.990892

[18] Jiri Hynek and Karel Jezek. 2003. Practical approach to automatic text summarization. In ELPUB. ELPUB, 1–11.

[19] Elena Baralis, Luca Cagliero, Saima Jabeen, and Alessandro Fiori. 2012. Multi-document summarization exploiting frequent itemsets. In Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, March 26-30, 2012. ACM, 782–786. DOI:http://dx.doi.org/10.1145/2245276.2245427

[20] Elena Maria Baralis, Luca Cagliero, Alessandro Fiori, and Saima Jabeen. 2011. PatTexSum: A pattern-based text summarizer. In Mining Complex Patterns Workshop. 18–29. Retrieved from http://porto.polito.it/2460874/.

[21] Elena Maria Baralis, Luca Cagliero, Alessandro Fiori, and Saima Jabeen. 2011. PatTexSum: A pattern-based text summarizer. In Mining Complex Patterns Workshop. 18–29. Retrieved from http://porto.polito.it/2460874/.

[22] Josef Steinberger, Mijail Kabadjov, Ralf Steinberger, Hristo Tanev, Marco Turchi, and Vanni Zavarella. 2011. JRC's participation at TAC 2011: Guided and multilingual summarization tasks. In TAC'11: Proceedings of the 2011 Text Analysis Conference. NIST, 1–9.

[23] Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. 2011. Text summarization using latent semantic analysis. J. Inf. Sci. 37, 4 (Aug. 2011), 405–417. DOI:http://dx.doi.org/10.1177/0165551511408848

[24] Josef Steinberger and Karel Jezek. 2009. Update summarization based on latent semantic analysis. In Proceedings of the 12th International Conference on Text, Speech and Dialogue (TSD'09), Pilsen, Czech Republic, September 13-17, 2009. Springer- Verlag Berlin Heidelberg, 77–84. DOI:http://dx.doi.org/10.1007/978-3-642-04208-9_14

[25] Jiawei Han, Jain Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In SIGMOD'00, Dallas, TX (May 2000). ACM.

[26] George Giannakopoulos, Mahmoud El-Haj, Benoit Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. 2011. TAC2011 MultiLing pilot overview. In TAC 2011 Workshop. NIST, Gaithersburg, MD. Retrieved from http://users.iit. demokritos.gr/ggianna/Publications/MultiLingOverview.pdf.

[27] John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In Proceedings of the COLING/ACL on Main Conference Poster Sessions (COLING-ACL'06). Association for Computational Linguistics, Stroudsburg, PA, 152–159. Retrieved from http://dl.acm.org/citation.cfm? id=1273073.1273093

[28] George Giannakopoulos. 2013. Multi-document multilingual summarization and evaluation tracks in ACL 2013 MultiLing workshop. In Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization. Association for Computational Linguistics, 20–28. Retrieved from http://www.aclweb.org/anthology/W13-3103.

[29] Eugene Krapivin, Mark Last, and Marina Litvak. 2014. JRouge - Java ROUGE Implementation. Retrieved July 2018 from https://bitbucket.org/nocgod/jrouge/wiki/Home/.

[30] Peter A. Rankel, John M. Conroy, and Judith D. Schlesinger. 2012. Better metrics to automatically predict the quality of a text summary. Algorithms 5, 4 (2012), 398–420. DOI:http://dx.doi.org/10.3390/a5040398

[31] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean *Efficient Estimation of Word Representations in Vector Space.*

[32] Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Bernd Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/TUD summarization system at TAC 2009. In Proceedings of the Text Analysis Conference (TAC'09). Gaithersburg, MD.

[33] Dan Gillick, Benoit Favre, and Dilek Hakkani-Tur. 2008. The ICSI summarization system at TAC 2008. In Proceedings of the Text Analysis Conference (TAC'08). Gaithersburg, MD.

[34] Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. Information Processing and Management 40, 6 (2004), 919–938.

[35] Xiaojun Wan and Jianwu Yang. 2006. Improved affinity graph based multi-document summarization. In Proceedings of HLT-NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 181–184.

[36] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 404–411. Retrieved from http://www.aclweb.org/anthology/W04-3252.

[37] Günes Erkan and Dragomir R. Radev. 2004. LexPageRank: Prestige in multi-document text summarization. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 365–371. Retrieved from http://www.aclweb.org/anthology/W04-3247.

[38] Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Bernd Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/TUD summarization system at TAC 2009. In Proceedings of the Text Analysis Conference (TAC'09). Gaithersburg, MD.

[39] Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI'12). AUAI Press, Arlington, VA, 479–490. Retrieved from

http://dl.acm.org/citation.cfm?id=3020652.3020704
[40] Xiaojun Wan and Jianwu Yang. 2006. Improved affinity graph based multi-document summarization. In Proceedings of HLT-NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 181–184.
[41] N. Rotem. 2011. Open text summarizer (OTS). Retrieved July 2017 from http://libots.sourceforge.net/.
[42] TexLexAn. 2011. TexLexAn: An open-source text summarizer. Retrieved July 2017 from http://texlexan.sourceforge.net/.
[43] Josef Steinberger. 2013. The UWB summariser at MultiLing-2013. In Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization. Association for Computational Linguistics, 50–54. Retrieved from http://www.aclweb.org/anthology/W13-3107.
[44] Josef Steinberger and Karel Ježek. 2005. Text Summarization and Singular Value Decomposition. Springer, Berlin, Berlin, 245–254. DOI:http://dx.doi.org/10.1007/978-3-540-30198-1_25
[45] Merijn van Erp and Lambert Schomaker. 2000. Variants of the Borda countmethod for combining ranked classifier hypotheses. In 7th International Workshop on Frontiers in Handwriting Recognition. International Unipen Foundation, 443–452.
[46] Unsupervised Learning of Sentence Embeddingsusing Compositional n-Gram Features Matteo Pagliardini, Iprova SA, Switzerland, Prakhar Gupta, EPFL, Martin Jaggi, EPFL.
[47] Michael Mampaey, Nikolaj Tatti, and Jilles Vreeken. 2011. Tell me what I need to know: Succinctly summarizing data with itemsets. In Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 573–581.