Central embeddings for extractive summarization based on similarity

Sandra J. Gutiérrez-Hinojosa, Hiram Calvo and Marco A. Moreno-Armendáriz
Centro de Investigación en Computación, Instituto Politécnico Nacional
J.D. Bátiz e/ M.O. de Mendizábal, 07738, Mexico City, Mexico
E-mail: b160499@sagitario.cic.ipn.mx, {hcalvo,mam_armendariz}@cic.ipn.mx

***Abstract*—In this work we propose using word embeddings combined with unsupervised methods such as clustering for the multi-document summarization task of DUC (Document Understanding Conference) 2002. We aim to find evidence that semantic information is kept in word embeddings and this representation is subject to be grouped based on their similarity, so that main ideas can be identified in sets of documents. We experiment with different clustering methods to extract candidates for the multi-document summarization task. Our experiments show that our method is able to find the prevalent ideas. ROUGE measures of our experiments are similar to the state of the art, despite the fact that not all the main ideas are found; as our method does not require annotated resources, it provides a domain and language independent way to create a summary.**

Keywords. Extractive Summarization, Prevalent Ideas Extraction, Concept Similarity, Central Embeddings, DUC 2002.

I. Introduction

Automatic summarization is a challenging task, as there are many issues such as redundancy, temporal handling, co-reference, sentence order, etc. that need particular attention when summarizing multiple documents, thereby, making this task complex (Gupta and Lehal, 2010).

A summary contains the main ideas of documents; in order to perform this task automatically, there are two different approaches: paraphrasing the main ideas of a document, or extracting sentences from the documents representing main ideas. This work focuses on this latter approach, called extractive summarization. The purpose of an algorithm for text summarization is to create a document formed by the most relevant information (Gambhir and Gupta, 2017); even for humans, this is a crucial step. There are several ways to determine which sentences are the most relevant in a set of documents.

Many algorithms to extract salient sentences from texts have been developed since the 1950s, when automatic text summarization arose. The first algorithm was based on topic representation, based on the idea that the more often a word repeats, the more likely it is to be important for identifying in the document (Luhn, 1958). This representation does not capture semantic and syntactic information; nevertheless, recent works with a similar approach have had a performance of 48% (recall) in a well-known dataset such as DUC (Document Understanding Conferences) 2002 (Wang and Li, 2012). Combining topic representation (word space models) with syntactic information such as Part Of Speech (POS) tagging helps to improve performance up to 55% (recall) (John et al., 2017).

In this work we propose using word embeddings combined with unsupervised clustering for the multi-document summarization task of DUC 2002. We aim to find evidence that semantic information is kept in word embeddings and this representation is subject to be grouped based on their similarity, so that main ideas can be identified in documents.

The following subsection describes related work to the task of document summarization, then in Section II we give some preliminaries related to this work. Our proposal is detailed in Section III, Results are discussed in Section IV, and finally conclusions are drawn in Section V.

A. Related works

Word embedding is a distributed vector representation technique to capture information of a word. Each column of the vector represents a latent feature of the word and captures useful semantic properties Mikolov et al. (2013). This representation obtains good performance of 53% (recall) for the summarization task maximizing a submodular function defined by the sum of cosine similarities based on sentence embeddings Kågebäck et al. (2014) and 56% (recall) using an objective function defined by a cosine similarity based on document embeddings. This function is calculated based on the nearest neighbors distances on embedding distributions Kobayashi et al. (2015). These works show that word embeddings are a useful representation to obtain the main ideas in the documents, but rely on the definition of an objetive function adjusted to a particular domain.

The main stages to obtain an extractive summary are three: representation, scoring, and selection. The representation contains the relevant features of the text; in the scoring stage each sentence obtains a weight using a similarity metric and finally, in the selection stage the summary length constraint is satisfied.

In the original DUC 2002 competition, ten algorithms were submitted for the extractive summarization task (200 words length). Halteren (2002) used weighted sentence scoring based

This research was funded by CONACyT-SNI and Instituto Politecnico Nacional (IPN), particularly through grants SIP20195886, EDI, and COFAA-SIBE

on lexical content. This method scores sentences with higher values when the sentence is different from the others.

Some other techniques are topic-based, such as Latent Semantic Allocation (LSA), a probabilistic method that extracts semantic structure in the text that uses the document context for extracts information about word relations. A higher number of common words among sentences indicate that the sentences are semantically related. Another technique called Singular Value Decomposition (SVD) is a linear algebra method which finds the interrelations between sentences and words using matrix representation.

Wang et al. proposed a Bayesian sentence-based topic model by using the term-document and the term-sentences matrices; each row represents a term and each column represents the document and the sentences, respectively. The goal of topic models is to infer words related to a topic and the topics discussed in a document. A higher value in each location indicates that the sentence or document is strongly related to that term (Wang et al., 2009).

The centroid-based method (Radev et al., 2004) is one of the most popular extractive summarization methods; it generates summaries using cluster centroids produced by topic detection, i.e. assesses the centrality of each sentence in a cluster and extracts the most important one.

A centroid is a set of words that are statistically important for the document cluster; therefore, the centroids are used to classify relevant documents and to identify salient sentences in a cluster. Each document is represented as a weighted vector of TF·IDF and the centroid is calculated using the first document. As new documents are processed, the TF·IDF values are compared with the centroid using cosine similarity, if the similarity is within a threshold, the new document is included in the cluster. The hypothesis of Radev et al. is that sentences containing words from the centroid are indicative of the topic of the cluster; the obtained results prove this hypothesis. However, the used word representation (TF·IDF) does not fully capture the semantic information of the words (Radev et al., 2004)

Formulating the summarization task as an optimization problem defines objective functions to evaluate candidate summaries. Objective functions are defined as essential parameters that a summary must accomplish, for example, coverage of all the main ideas John et al. (2017). Methods based on optimization methods have achieved best performance tested on DUC 2002. In Table I a summary of the best results is shown. A disadvantage of establishing objective functions is that they are adjusted based on a particular document set or domain, and thus, they might not represent a general way of creating summaries. This is why in this work we explore different ways of creating summaries, based on unsupervised clustering.

II. Preliminaries

In this section, details on the task in general (Section II-A) are presented. Then we discuss some text preprocessing techniques (Section II-B), and evaluation methods (Section II-C). The word embeddings used in this work are described in Section II-D, along with the used similarity measures (Section II-E).

A. The summarization task

The main goal of a summary is to encompass the main ideas in a document reducing the original document size. If all sentences in a text document were of equal importance, producing a summary would not be very effective, as any reduction in the size of a document would carry a proportional decrease in its informativeness. However, identifying the most relevant segments in the documents is the main challenge in summarization.

This task produces a transformation of source documents through content condensation by selecting and generalizing on important information (Jones, 2007). Also, algorithms created for solving this task have a relevant application given the exponential growth of textual information online, and the need to find the main ideas of documents in a shorter time.

Research on the summarization task started to attract the attention of the scientific community in the late fifties when there was a particular interest in the automation of summarization for the production of abstracts of technical documentation Luhn (1958).

1) Summarization types: There are several distinctions in summarization, some are described below:

1) Source type: **single-document**, where a summary of a single document is produced; whereas in **multi-document** a summary of many documents on the same topic or the same event is built.
2) Output produced: **extractive**, which is a summary containing passages selected from the source document (usually sentences); and **abstractive**, where the information from the source document has been analyzed and transformed using paraphrasing, reorganizing, modifying and merging information for condensation.
3) Language: **mono-lingual**, where the language of the source document is the same for the summary; **multi-lingual**, which accepts two or more languages from a source document; and **cross-lingual**, which translates the summary to other than the original language.
4) Audience-oriented: **generic**, in this output it is assumed that anyone may end up reading the summary; and **query-oriented** that provides a summary that is relevant to a specific user query.

This work focuses on a multi-document, extractive, mono-lingual (English) and generic summary.

B. Text preprocessing techniques

Some of the most used techniques are:

- Word and sentence tokenization: Tokenization is the process of separating the text into words, phrases, symbols, or other meaningful elements called tokens. This process is considered easy compared to other tasks in natural language. However, automatically extracted text may contain inaccurately compounded tokens, spelling errors and unexpected characters that can be propagated into later

TABLE I
COMPARISON OF RECALL METRICS FOR SUMMARIES.

Work	Method	ROUGE-1	ROUGE-2	F-score	Dataset
Halteren (2002)	Scoring based on lexical content	0.2000	-	0.2100	DUC 2002
Radev et al. (2004a)	Centroid-cluster	0.4538	0.1918	-	Extracted by CDIR
Wang et al. (2009)	Position, semantic, LSA-NMF	0.4881	0.2457	-	DUC 2002
John et al. (2017)	Optimization	0.5532	0.2586	0.5419	DUC 2002

phases causing problems. Therefore, tokenization is an important step and in some cases needs to be customized to the data in question.

In all modern languages that are based on Latin, Cyrillic, or Greek classical languages, such as English, word tokens are delimited by a blank space. In these languages, also called segmented languages, token boundary identification is not a complex task, an algorithm which replaces white spaces with word boundaries and inserts a white space when a word is followed by a punctuation mark will produce a reasonable performance. Nonetheless, a period is an ambiguous punctuation mark indicating a full-stop, a part of abbreviation or both. Regular expressions can resolve these ambiguities defining different string search patterns Thompson (1968). In this work Python libraries have been used for tokenization[1]

- Stop words removal: The stop words are common and non-informative words that are often filtered, such as articles, prepositions, pronouns, etc. The removal of stop words have been done using methods based on Zipf's law (Zipf, 1949), these methods indicate a distribution of words for any corpus and established an upper and lower cut-off frequency, being stop words the ones that are not between the cut-off.

  Analyzing a dataset shows the most frequent words are document type dependent. A definitive stop words list does not exist, therefore the list used in this work is a general one[2] and contains 153 items.
- Stemming: This method is used to reduce words to a common form by removing their longest ending handling spelling exceptions. Two main principles are used in the construction of a stemming algorithm: iteration and longest-match. Iteration is based on the fact that suffixes are attached to stems in a certain order, no more than one match is allowed within a single order-class; and the longest-match principle states that within any given class of endings, the longest ending should be removed.

  The stemming algorithm[3] used in this work is based on (Porter, 2001).

C. Summary evaluation

There are two quality evaluation methods for the summarization task: extrinsic and intrinsic. Extrinsic methods are based on the performance of a specific task (question-answering, comprehension, etc.) while intrinsic measures are based on norm set (fluency, coverage, similarity to an annotator summary, etc.).

Both quality evaluation methods can be performed by a human or a machine. The automatic evaluation lacks the linguistic skills and emotional perspective that a human has, but is popular because the evaluation process is quick, even when the summaries are large, and provides a consistent way of comparing the various summarization algorithms Fiori (2014).

1) Recall-Oriented Understudy for Gisting Evaluation (ROUGE): The University of Southern California's Information Sciences Institute (ISI) developed the recall-based metric called ROUGE-N defined by Equation 1 (Lin and Hovy, 2003).

$$ROUGE\text{-}N = \frac{\sum_{s \in GSS} \sum_{n\text{-}gram} Count_{match}(n\text{-}gram)}{\sum_{s \in GSS} \sum_{n\text{-}gram} Count(n\text{-}gram)} \quad (1)$$

Where N is the number of *n*-grams, GSS is a set formed by the gold-standard summaries s, $Count_{match}(n\text{-}gram)$ is the number of *n*-grams co-occurring in gold-standard and the retrieved summaries and $Count(n\text{-}gram)$ is the number of *n*-grams in the gold-standard summary (Lin and Hovy, 2003).

In 2004, the ROUGE package was created including additional recall-based metrics, such as ROUGE-L, ROUGE-W, ROUGE-S, etc. and their precision and F-score metrics[4]. This package has a maximum reference count, i.e. if the word is repeated it only counts the number of times it is repeated in the gold-standard summary.

ROUGE metrics were evaluated to measure their correlation with human evaluations; for the multi-document summarization task, ROUGE-1 and ROUGE-2 showed high Pearson's correlation (90%) (Lin, 2004). These two metrics are used in this work.

2) Document Understanding Conferences: In 2000, to foster progress in summarization, and as a part of an evaluation campaign organized by the Defense Advanced Research Projects Administration (DARPA) and the National Institute of Standards and Technology (NIST), the Document Understanding Conferences (DUC) were created[5]. In these challenges, different summarization tasks were developed by NIST and data for training and testing was distributed to participants.

In the dataset of DUC 2002 for the multi-document extractive summarization task (200 words length) 60 collections (document sets) with two gold-standard summaries each were distributed, but due to reasons beyond our knowledge, one document collection (d088) received no gold-standard summaries and two collections (d076 and d098) received only

[1] https://www.nltk.org/_modules/nltk/tokenize.html
[2] based on http://snowball.tartarus.org/algorithms/english/stop.txt
[3] http://www.nltk.org/_modules/nltk/stem/snowball.html#EnglishStemmer
[4] http://rxnlp.com/rouge-2-0/
[5] https://www-nlpir.nist.gov/projects/duc/intro.html

one; therefore, only 57 collections have the two gold-standard summaries. In this work, the dataset of DUC 2002 with 57 collections was used (Halteren, 2002).

D. Word embeddings

The term word embedding was originally coined by Bengio et al. (2003). They trained a neural network to predict the next word given previous words in order to obtain a feature vector associated with each word; similar words are expected to have similar feature vectors. However, Collobert and Weston (2008) were the first to demonstrate the power of pre-trained word embeddings and establish word embeddings as a highly effective tool when used in natural language processing tasks. Moreover, in (Mikolov et al., 2013) word embeddings were brought to the fore through the creation of word2Vec and a tool-kit enabling the training and use of pre-trained embeddings.

Word2Vec is an efficient method for learning high-quality vector representations of words from large amounts of text data using neural networks. There are two models for computing embeddings: the bag-of-words and skip-gram models. Bag-of-words model predicts the probability of a word given a context, while the skip-gram model predicts the context given a word (Mikolov et al., 2013).

Word embeddings result from applying unsupervised learning, therefore they do not require annotated datasets. Rather, they can be derived from already available unannotated corpora.

1) Paragraph embeddings: With the success of word embeddings, new algorithms called paragraph embeddings were developed. These paragraph embeddings, based on word embeddings, are an unsupervised learning algorithm that learns vector representations for variable length pieces of texts such as sentences and documents. As in word embeddings, there are two models: memory model and bag-of-words. Memory model predicts a paragraph identification given a number of context words, while the bag of words model ignores context words and forces the model to predict words randomly sampled from the paragraph in the output layer (Le and Mikolov, 2014).

A software framework implementing these techniques was created (Rehurek and Sojka, 2010) and the method was named doc2Vec[6]. In (Lau and Baldwin, 2016) they performed an empirical evaluation of doc2Vec on two tasks: duplicate question detection in a web forum and semantic textual similarity between two sentences, finding that doc2Vec in bag-of-words model performs better than the memory model. In this work, the final hyper-parameters and model of the previous work have been used[7].

E. Similarity measure

Measuring similarity between vectors is related to measuring the distance between them, the smaller the distance the larger the similarity.

Finding similarity between words is a fundamental part of finding the sentence, paragraph and document similarities. Words can be similar in two ways: lexically and semantically. Words are similar lexically if they have a similar character sequence. Words are similar semantically if they are used in the same context (Gomaa and Fahmy, 2013).

Word and paragraph embeddings are a representation that contains lexical and semantic information in vector form, therefore to measuring their similarity vector distance has to be computed.

Cosine similarity measures the distance between two vectors using an inner product that measures the angle between them, as shown in Equation 2.

$$D = \frac{x \cdot y}{\parallel x \parallel \parallel y \parallel} \tag{2}$$

Euclidean distance is the square root of the sum of squared differences between corresponding elements of the two vectors, also called L2-distance, as shown in Equation 3.

$$D = \sum_{i}^{N} \sqrt{(X_i - X_j)^2} \tag{3}$$

Where N is the dimension of each vector and i and j are the two vectors.

III. Proposal

The proposed method for the multi-document summarization task consists of three stages: (a) pre-process the dataset DUC 2002 in order to eliminate non-content words (Section III-A) (b) select a word representation for this dataset to capture semantic and syntactic information and obtain sentence vectors (Section III-B); and (c) implement a method to obtain the main ideas of the documents and select the relevant ones (Section III-C). The sentences with main ideas will form the summary; this is the general approach to generate a summary, as described in Section II-A. In Figure 1 a general diagram of the proposal is shown.

A. Pre-processing stage

In the first stage sentence tokenization and word tokenization of each sentence are implemented using Python libraries based on regular expressions and named entities recognition[8].

The DUC 2002 dataset has 547 documents D grouped in 57 document sets T with two gold-standard summaries each.

In this work, each set is tokenized in sentences and each sentence is tokenized in words; then the stop words have been removed using a list of non-informative words for the English language[9] and stemming of each word in a sentence has been done, based on Porter's stemming[10], this is shown in Figure 2.

[6]https://radimrehurek.com/gensim/models/doc2vec.html
[7]https://github.com/jhlau/doc2vec

[8]https://www.nltk.org/_modules/nltk/tokenize.html
[9]http://snowball.tartarus.org/algorithms/english/stop.txt
[10]http://www.nltk.org/_modules/nltk/stem/snowball.html#EnglishStemmer

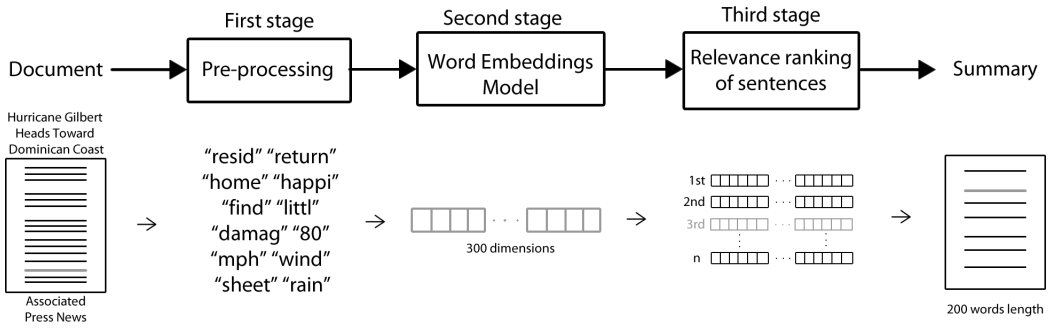


Fig. 1. General scheme for the proposal.

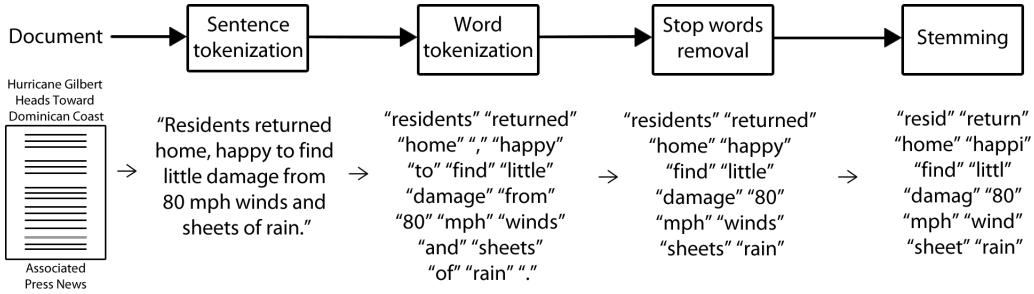


Fig. 2. Pre-processing stage.

B. Embeddings model

In the second stage, the word embedding model *doc2Vec* is used[11]. The model is a trained Artificial Neural Network (ANN) whose input is a set of tokenized words—which can be extracted from a sentence, a document, or a set of documents—and its output is a vector of 300 dimensions, called, from now on, simply *embedding*, i.e., an embedding is a vector representation of the sentence, document, or set of documents.

An example of the embedding model used in this work is shown in Figure 4, where the input is a sentence of the DUC dataset and the output is an embedding. The final hyper-parameters and model described in (Lau and Baldwin, 2016) have been used.

In order to find the sentence that has been transformed to a vector, let $S_i^{j,t}$ be the i-th sentence belonging to the j-th document of the t set of documents, and $E(S_i^{j,t})$ the embedding corresponding to $S_i^{j,t}$. Then we define a function f that associates each element in S set with an element in E(S), as described by Equation 4.

$$f : E(S_i^{j,t}) \rightarrow S_i^{j,t} \tag{4}$$

This function keeps an index to map a vector to the sentence that originated it (one to one function) in order to build the summary with the corresponding sentence from specific embeddings.

C. Relevance ranking of the sentences

In the last stage, once the embeddings for each sentence in the DUC dataset are obtained, we propose to calculate a central vector that contains the document main ideas, using the average of the sentence embeddings.

For this approach, we consider that each column of the sentence embedding represents a document subject and a higher value indicates the important subjects. Consequently, the average of the sentence embeddings in a set should represent the subjects in the set and higher value columns indicate the important subjects in the set; we call this average *central embedding* (CE) and consider that it represents the main idea of the document set.

The distance between each paragraph embedding and the central embedding in each column is short if both match in most columns; this means that the sentence embedding contains the same subjects than the main idea (central embedding) of the document set, and thus it is a relevant embedding that must be included in the summary; an example of this is shown in Figure 3.

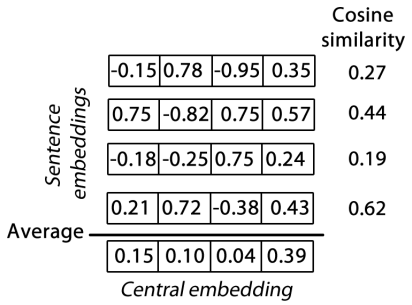


Fig. 3. Example of averaging vectors.

In this toy example, four vectors with four columns each are shown. The average of each column is calculated, this vector is the central embedding and the cosine similarity with

[11] https://github.com/jhlau/doc2vec

each sentence embedding is shown with the aim to show that the last sentence embedding has a higher similarity because the distance in each column is shorter than the other sentence embeddings.

Recalling that word embedding model input, shown in Figure 4, could be a word, a sentence, a paragraph, a document or text of any length. We propose three different forms of computing the central embedding: (a) using the sentence embeddings (**CE-S**), as in Figure 3; (b) using the document embeddings (**CE-D**), instead of using the sentence embeddings in Figure 3 (i.e., the input in Figure 4 is a document); and (c) the central embedding is the document set embedding (**CE-Set**), this means that the input in Figure 4 is a document set. In Figure 5 these variants are illustrated. Algorithm 1 shows the pseudo-code for computing these three variants.

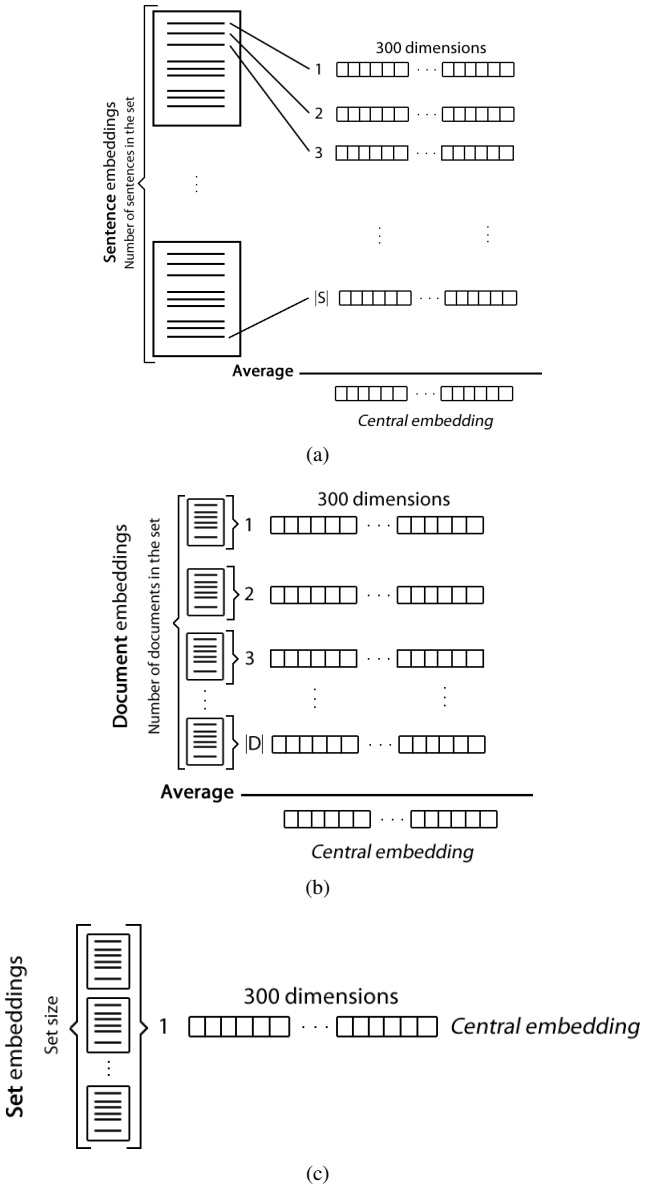


Fig. 5. Different ways of computing average vectors: (a) CE computed using the sentences (**CE-S**); (b) CE computed using the documents (**CE-D**) and; (c) CE using the document set (**CE-Set**). $|S|$ is the number of sentences in each set, and $|D|$ is the number of documents in each set.

Once the central embedding is obtained, the cosine similarity between each sentence embedding and the calculated central embedding is calculated, giving ranked sentence embeddings related to a sentence using Equation 4. The top ranked sentences will form the summary. This process is depicted in Figure 6 and detailed in Algorithm 2. The central embedding CE can be CE-S, CE-D or CE-Set, as previously calculated.

IV. Results

For the evaluation of the summaries in each experiment presented in Section III, two measures have been used: ROUGE-1, ROUGE-2, and F-score because they have a high correlation with human evaluation of summaries (Lin, 2004).

Recalling that each document set contains two gold-standard summaries of 200 words length, we present the results comparing each gold-standard summary separately, and using both as an average result.

Three different forms of computing the central embedding were proposed: (a) using the sentences (**CE-S**), (b) using the documents (**CE-D**) and (c) using the document set (**CE-Set**). In Tables II, III and IV ROUGE-1, ROUGE-2 and F-score for the experiments are shown.

TABLE II
Recall for CE-S experiment.

Gold-standard summary set	ROUGE-1	ROUGE-2	F-score
A	0.3808	0.1087	0.3740
B	0.3671	0.0932	0.3605
A and B	0.3740	0.1010	0.3673

TABLE III
Recall for CE-D experiment.

Gold-standard summary set	ROUGE-1	ROUGE-2	F-score
A	0.4371	0.1906	0.4495
B	0.4336	0.1873	0.4454
A and B	**0.4353**	**0.1889**	**0.4474**

TABLE IV
Recall for CE-Set experiment.

Gold-standard summary set	ROUGE-1	ROUGE-2	F-score
A	0.4233	0.1791	0.4365
B	0.4119	0.1633	0.4239
A and B	0.4176	0.1712	0.4302

Our calculated embeddings for the DUC 2002 dataset, as well as the different Central Embeddings are available online[12]

A. Case study of a document set

The main hypothesis of this proposal is that a central embedding should contain the main idea of a document and therefore, the sentence embeddings close to this central embedding contain important ideas because of their similarity.

[12]http://dx.doi.org/10.21227/qq4m-er38

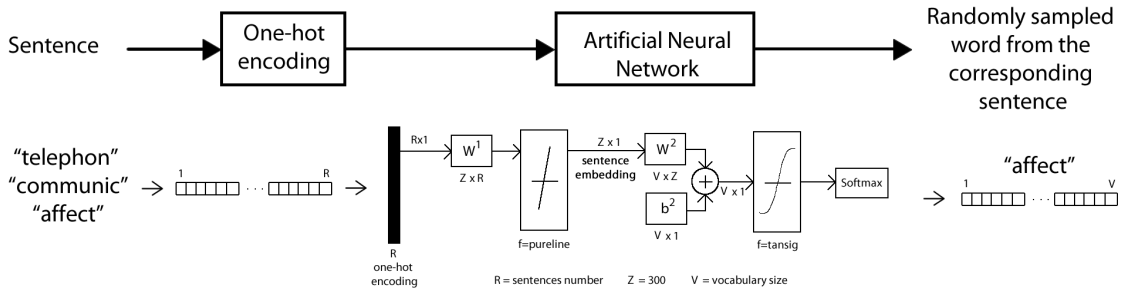


Fig. 4. The artificial neural network architecture used to train (Lau and Baldwin, 2016).

Algorithm 1 Central Embeddings calculation

procedure CE(T^t)
input: T^t, a set of documents to summarize.
output: $CES, CED, CESet$, central embedding vectors per sentence, document and set.
    Let $D^{j,t}$ be the j-th document belonging to the t set of documents.
    Let $S_i^{j,t}$ be the i-th sentence belonging to the j-th document of the t set of documents.
    Let $E(S_i^{j,t})$ be the embedding corresponding to $S_i^{j,t}$.
    Let $E(D^{j,t})$ be the embedding corresponding to $D^{j,t}$.
    Let $E(T^t)$ be the embedding corresponding to all documents of the t set.
    $nsent_t \leftarrow |\bigcup_{i,j} S_i^{j,t}|$ ▷ the total number of sentences for a given set t
    $ndoc_t \leftarrow |\bigcup_{j} D_i^{j,t}|$ ▷ the total number of documents for a given set t
    $CES_t = \sum_{i,j} \frac{E(S_i^{j,t})}{nsent_t}$ ▷ Central Embedding per sentence
    $CED_t = \sum_{j} \frac{E(D^{j,t})}{ndoc_t}$ ▷ Central Embedding per document
    $CESet_t = E(T^t)$ ▷ Central Embedding per set
    return $CES, CED, CESet$
end procedure

Algorithm 2 Sentence selection

procedure SELECT($T^t, CE, length$)
input: T^t, a set of documents to summarize, CE a central embedding, $length$ (in words) of the summary
output: a summary of length $length$.
    $R_i^{j,t} \leftarrow sim_{cos}(E(S_i^{j,t}), CE) \quad \forall i, j$ ▷ cosine similarity
    $rSim \leftarrow rank_{top-down}(R_i^{j,t})$ ▷ create a list of embeddings ordered from most- to less-similar to CE
    $nwords \leftarrow 0$ ▷ number of words
    for each e in $rSim$ **do** ▷ e is an embedding
        find $S_i^{j,t}$ corresponding to e using eq. 4
        print $S_i^{j,t}$
        $nwords \leftarrow nwords + |S_i^{j,t}|$
        if $(nwords > length)$ **then**
            return
        end if
    end for
end procedure

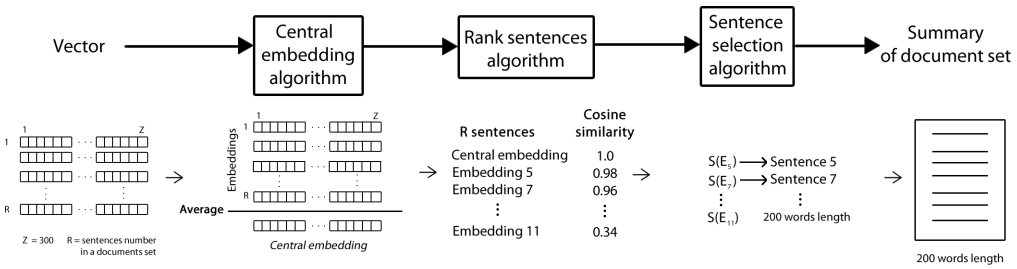


Fig. 6. Procedure to find relevant main ideas of documents set using Central Embeddings (CE)

We will examine a particular document set, composed of 6 documents ($|D^t| = 6$) and 480 sentences ($|S^t| = 488$) dealing with the event of Hurricane Hilbert moving towards the coast. Different Associated Press reports from different countries (Jamaica, Santo Domingo, Mexico (Yucatan), USA (Miami)), and a Wall Street Journal article are included. Figure 7 shows the provided abstracts by two different annotators (Gold standard A and B). We have marked sentences according to the subject they cover. In yellow, sentences dealing with the hurricane's route are highlighted. Those reporting damages are marked in green, and orange highlights sentences dealing with hurricane's characteristic features. It can be seen that both summaries contain these three subjects in a balanced way.

Generated summaries of our system on this set of documents are shown in Figure 8. The first strategy (CE-S) includes several sentences (7, 10, 11, 12) that are not clearly identified under the three previously mentioned main subjects; few sentences are included on the damage report. The second strategy (CE-D, Figure 8(b)) gives the best balance on the three subjects mentioned. The third strategy (CE-Set, Figure 8(c)) lacks information on the route description. Although more descriptive sentences are obtained with the CE-S experiment, the best performance was obtained in the CE-D experiment because in CE-S the sentence descriptions are not related with route and origin of the hurricane.

The effect of locating the central embedding with different strategies can be observed in Figure 9. These plots were created using the Radviz library[13], which allows to project the 300-dimension embeddings into a 2D plot for visualization purposes. Circle markers indicate the document set sentences; triangle markers indicate the selected sentences; square markers indicate the location of the central embedding; cross markers indicate the central embeddings for the A gold standard, while star markers indicate the central embeddings for the B gold standard.

In Figure 9(a) the central embedding is located in the center of the selected sentences and very close to gold-standard central embeddings, but the sentences are short and do not contain relevant topics; this summary only selects one sentence from each gold-standard. In Figure 9(b) the central embedding seems far from the gold-standard central embeddings but contains more information from damage reports; it has two sentences in common with the gold-standard summaries. In Figure 9(c) the central embedding is far from the gold-standard central embeddings, but is close to the central embedding of the previous experiment containing sentences with the damage report. It has five sentences in common with the previous experiment and two sentences in common with the gold-standard summaries.

[13]https://cran.r-project.org/web/packages/Radviz/vignettes/single_cell_projections.html

V. Conclusions

In this paper we addressed the multi-document summarization task using a word embedding model that represents sentences as vectors which contain syntactic and semantic information.

Different variants to calculate central embeddings have been described. Specifically, three different ways of calculating averages were proposed: (1) using the sentences, (2) using the documents and (3) using the document set. Their foundations rely on the centroid-based method, which indicates that sentences containing words from the centroid are more indicative of the document topic. We found that using the documents for calculating the averages yielded better results when evaluated on the DUC 2002 corpus.

Our method obtains similar performance to LSA methods with the advantages that sentence embeddings do not have the course of dimensionality of the matrices and are independent of the document type and language. This implies that sentence embeddings obtain semantic information useful for summaries and the results could be improved if different main ideas can be found in sentence groups, for example, sentences with a predominant description of the origin and route of the hurricane.

The advantages of this method are: it does not need any linguistic resource, it is easy to implement and has a similar performance to the state of the art. Also, our method is unsupervised, thus it can be adapted to other summarization corpora and language without the need of adjusting parameters, or estimating optimization goals.

In future work, we plan to evaluate the results with clustering algorithms in order to obtain different groups of main ideas in order to capture the balance of topics observed in gold-standard summaries. Also, testing our method on different corpora to evaluate its performance is left as future work.

VI. Acknowledgments

The authors wish to thank the Government of Mexico (Instituto Politécnico Nacional, SNI, SIP-IPN, COFAA-IPN, BEIFI-IPN and CONACyT) for providing necessary support to carry out this research work.

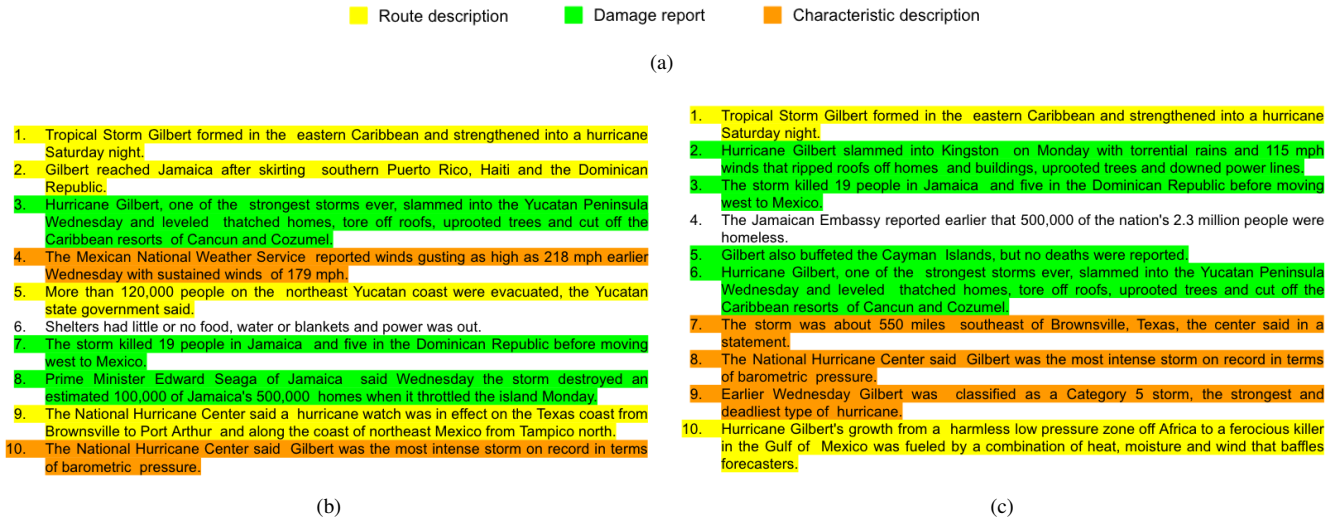


Fig. 7. Gold standard summaries from DUC 2002 (document set d061j) (a) legend; (b) Gold Standard A and; (c) Gold Standard B.

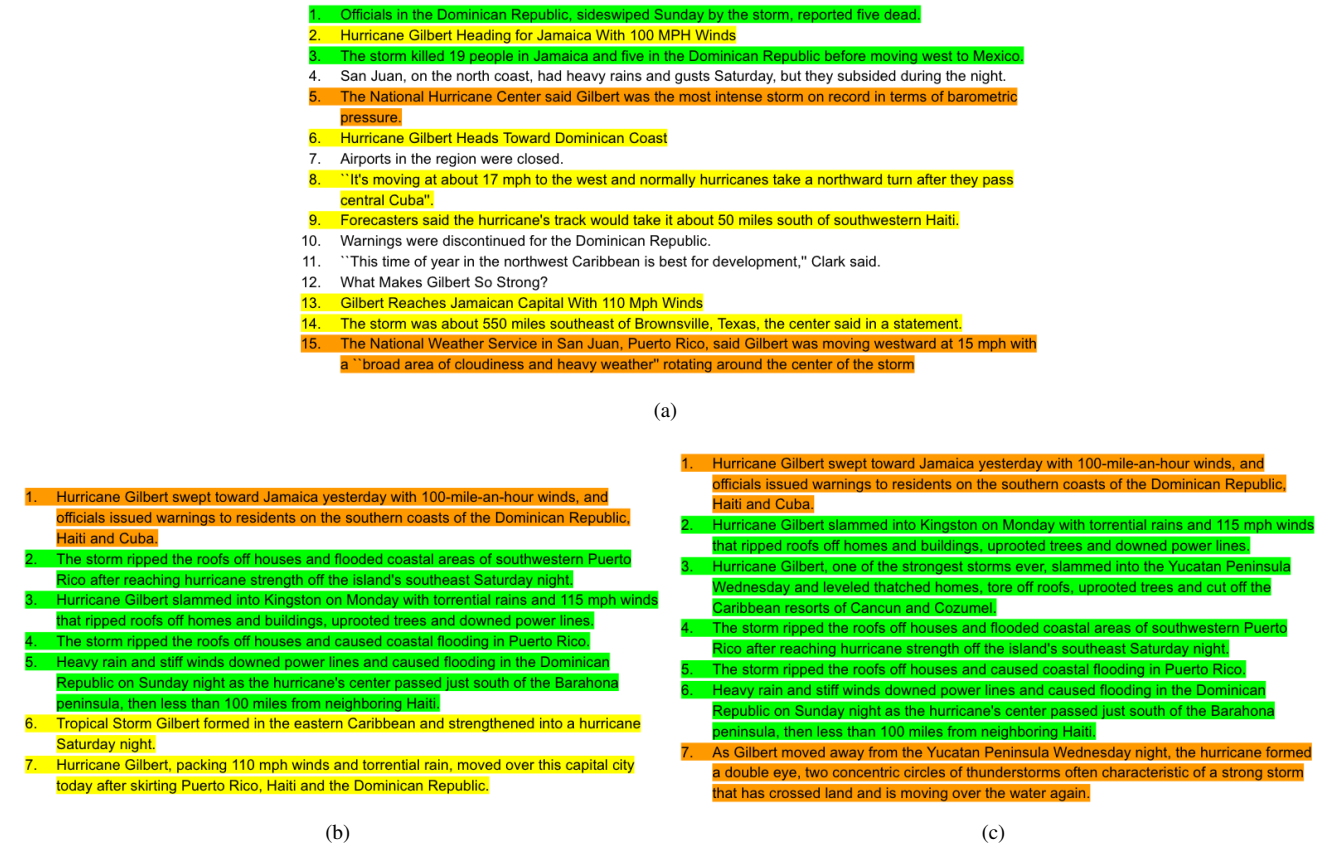
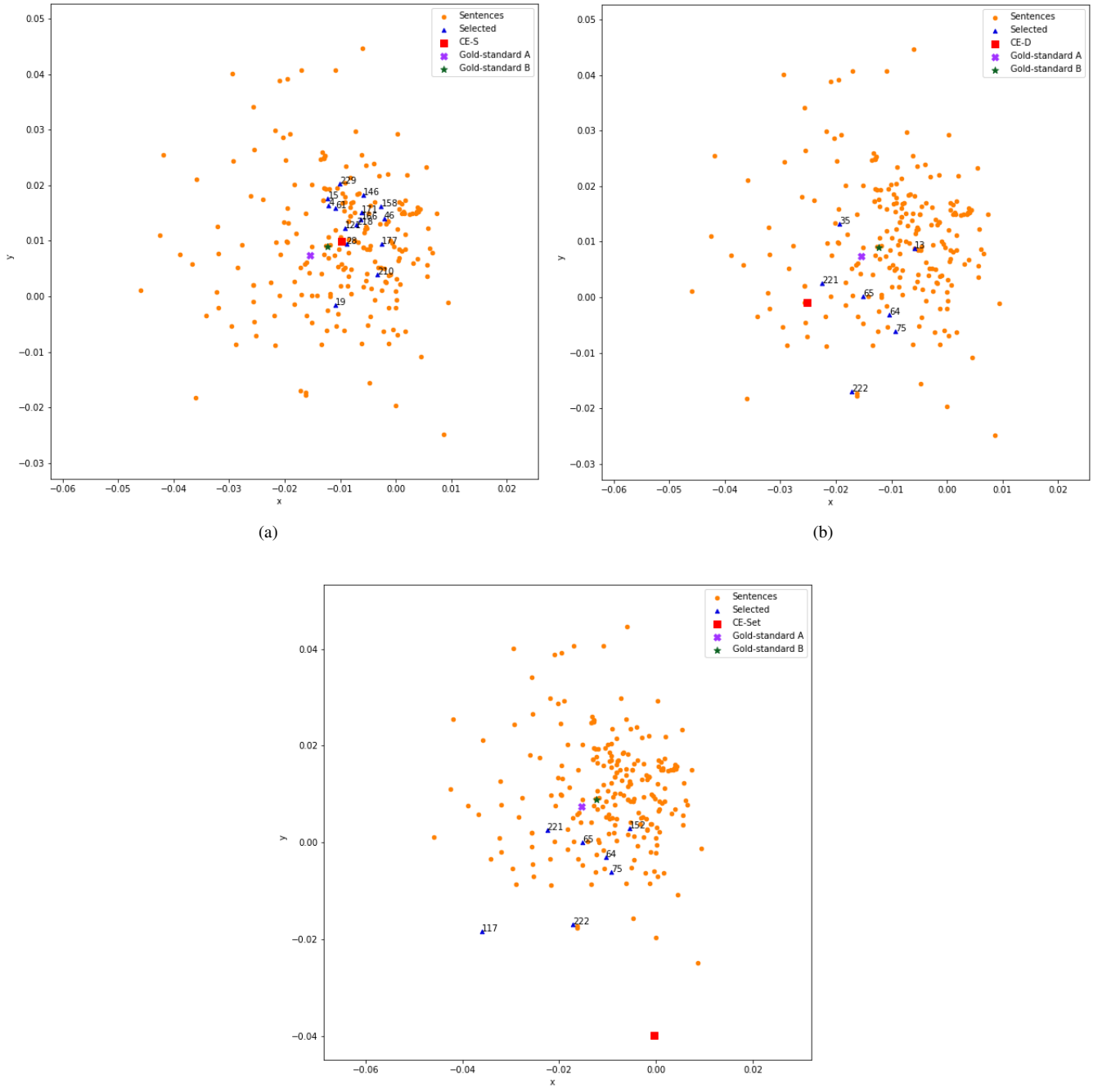


Fig. 8. Generated summaries (for DUC 2002 document set d061j) (a) CE-S; (b) CE-D; (c) CE-Set;



(a) (b)

(c)

Fig. 9. Visualization of the different forms of computing central embeddings for the d061j document set: (a) CE calculated using the sentences (**CE-S**); (b) CE calculated using the documents (**CE-D**) and; (c) CE using the document set (**CE-Set**).

REFERENCES

V. Gupta, G. S. Lehal, A survey of text summarization extractive techniques, Journal of emerging technologies in web intelligence 2 (2010) 258–268.

M. Gambhir, V. Gupta, Recent automatic text summarization techniques: a survey, Artificial Intelligence Review (2017) 1–66.

H. P. Luhn, The automatic creation of literature abstracts, IBM Journal of research and development 2 (1958) 159–165.

D. Wang, T. Li, Weighted consensus multi-document summarization, Information Processing & Management 48 (2012) 513–523.

A. John, P. Premjith, M. Wilscy, Extractive multi-document summarization using population-based multicriteria optimization, Expert Systems with Applications 86 (2017) 385–397.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., USA, 2013, pp. 3111–3119.

M. Kågebäck, O. Mogren, N. Tahmasebi, D. Dubhashi, Extractive summarization using continuous vector space models, in: Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL, pp. 31–39.

H. Kobayashi, M. Noguchi, T. Yatsuka, Summarization based on embedding distributions, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1984–1989.

H. v. Halteren, Writing style recognition and sentence extraction (2002).

D. Wang, S. Zhu, T. Li, Y. Gong, Multi-document summarization using sentence-based topic models, in: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, Association for Computational Linguistics, pp. 297–300.

D. R. Radev, H. Jing, M. Styś, D. Tam, Centroid-based summarization of multiple documents, Information Processing & Management 40 (2004) 919–938.

K. S. Jones, Automatic summarising: The state of the art, Information Processing & Management 43 (2007) 1449–1481.

K. Thompson, Programming techniques: Regular expression search algorithm, Communications of the ACM 11 (1968) 419–422.

G. K. Zipf, Human behaviour and the principle of least-effort. cambridge ma edn, Reading: Addison-Wesley (1949).

M. F. Porter, Snowball: A language for stemming algorithms, 2001.

A. Fiori, Innovative document summarization techniques: Revolutionizing knowledge understanding, IGI Global, Philadelphia, 2014.

C.-Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics, pp. 71–78.

C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, Text Summarization Branches Out (2004).

Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of machine learning research 3 (2003) 1137–1155.

R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, ACM, pp. 160–167.

T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, pp. 1188–1196.

R. Rehurek, P. Sojka, Software framework for topic modelling with large corpora, in: In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Citeseer.

J. H. Lau, T. Baldwin, An empirical evaluation of doc2vec with practical insights into document embedding generation, arXiv preprint arXiv:1607.05368 (2016).

W. H. Gomaa, A. A. Fahmy, A survey of text similarity approaches, International Journal of Computer Applications 68 (2013).

C.-Y. Lin, Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?, in: NTCIR.