Automatic Text Summarization with a Reduced Vocabulary Using Continuous Space Vectors

Elvys Linhares Pontes, Stéphane Huet, Juan-Manuel Torres-Moreno, Andréa Carneiro Linhares

Automatic Text Summarization with a Reduced Vocabulary Using Continuous Space Vectors

Elvys Linhares Pontes*[1], Stéphane Huet[1], Juan-Manuel Torres-Moreno*[1,2], and Andréa Carneiro Linhares[3]

[1] LIA, Université d'Avignon et des Pays de Vaucluse, Avignon, France
[2] École Polytechnique de Montréal, Montréal, Canada
[3] Universidade Federal do Ceará, Sobral-CE, Brasil

Abstract. In this paper, we propose a new method that uses continuous vectors to map words to a reduced vocabulary, in the context of Automatic Text Summarization (ATS). This method is evaluated on the MultiLing corpus by the ROUGE evaluation measures with four ATS systems. Our experiments show that the reduced vocabulary improves the performance of state-of-the-art systems.

Keywords: Word Embedding, Text Summarization, Vocabulary Reduction

1 Introduction

Nowadays, the amount of daily generated information is so large that it cannot be manually analyzed. Automatic Text Summarization (ATS) aims at producing a condensed text document retaining the most important information from one or more documents; it can facilitate the search for reference texts and accelerate their understanding.

Different methodologies based on graphs, optimization, word frequency or word co-occurrence have been used to automatically create summaries [12]. In the last years, Continuous Space Vectors (CSVs) have been employed in several studies to evaluate the similarity between sentences and to improve the summary quality [1, 5, 10]. In this paper, we introduce a novel use of CSVs for summarization. The method searches for similar words in the continuous space and regards them as identical, which reduces the size of vocabulary. Then, it computes metrics in this vocabulary space seen as a discrete space, in order to select the most relevance sentences.

After a brief reminder on the implementation of neural networks to build CSVs in Section 2, Sections 3 and 4 describe the previous works that used CSVs for summarization and our method respectively. In Section 5, we present the evaluation of the proposed approach with four ATS systems. Our results show that the reduced vocabulary in a discrete space improves the performance of the state-of-the-art. Section 6 concludes with a summary and future work.

* This work was partially financed by the European Project CHISTERA-AMIS ANR-15-CHR2-0001.

2 Neural Networks and Continuous Space Vectors

Artificial Neural Networks (ANNs) have been successfully applied in diverse Natural Language Processing applications, such as language modeling [3, 8], speech recognition or automatic translation. An ANN is a system that interconnects neurons and organizes them in input, hidden and output layers. The interest of these models has recently been renewed with the use of deep learning which updates the weights of the hidden layers to build complex representations.

Mikolov et al. [8] developed a successful approach with the so-called Skip-gram model to build continuous word representations, i.e., word embeddings. Their model aims at predicting a word, basing its decision on other words in the same sentence. It uses a window to limit the number of words used; e.g. for a window of 5, the system classifies the word w from the 5 words before and the 5 words after it. Given a sequence of training words $w_1, w_2, w_3, ..., w_N$, the objective of the Skip-gram model is to maximize the average log probability:

$$\frac{1}{N}\sum_{t=1}^{N}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j} \mid w_t) \tag{1}$$

where c is the window size and N is the number of words in the training set.

3 Related Work

Several works have used word embeddings to measure sentence similarity, which is central to select sentences for text summarization. Kågebäck et al. [5] used continuous vectors as semantically aware representations of sentences (phrase embeddings) to calculate the similarity, and compared the performance of different types of CSVs for summarization. Balikas and Amini [1] also analyzed various word embedding representations to summarize texts in English, French and Greek languages. They proposed an autoencoder model to learn a language independent representation for multilingual sentences and determine the similarity between sentences. Phung and De Vine [10] used word embeddings to calculate the sentence similarity measures for the PageRank system to select the most significant sentences. They compare this PageRank version with other systems based on TF-IDF and different variants of Maximum Marginal Relevance (MMR).

Our methodology differs from the previously described methods because we reduce the vocabulary using a CSV-based similarity between words. This first step allows us to calculate more accurately the similarity and the relevance of sentences in a discrete space.

4 Reduced Vocabulary

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

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

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

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

Algorithm 1 Reduce vocabulary of *text*

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

5 Experiments and Results

The reduced vocabulary approach was evaluated with four different systems. The first simple system (named "base") generates an extract with the sentences that are the most similar to the document. The second system (MMR) produces a

summary based on the relevance and the redundancy of the sentences [2]. With the objective of analyzing different methodologies to calculate the relevance and the similarity of sentences (e.g. word co-occurrence, TF-ISF[4]...), we use two other systems: Sasi [11] and TextRank [7].

Pontes et al. [11] use Graph theory to create multi-document summaries by extraction. Their so-called Sasi system models a text as a graph whose vertices represent sentences and edges connect two similar sentences. Their approach employs TF-ISF to rank sentences and creates a stable set of the graph. The summary is made of the sentences belonging to this stable set.

TextRank [7] is an algorithm based on graphs to measure the sentence relevance. The system creates a weighted graph associated with the text. Two sentences can be seen as a process of recommendation to refer to other sentences in the text based on a shared common content. The system uses the Pagerank system to stabilize the graph. After the ranking algorithm is run on the graph, the top ranked sentences are selected for inclusion in the summary.

We used for our experiments the 2011 MultiLing corpus [4] to analyze the summary quality in the English and French languages. Each corpus has 10 topics, each containing 10 texts. We concatenated the 10 texts of each topic to convert multiple documents into a single text. There are between 2 and 3 summaries created by human (reference summaries) for each topic. We took the LDC Gigaword corpus (5th edition for English, 3rd edition for French) and the word2vec package[5] to create the word embedding representation, the vector dimension parameter having been set to 300. We varied the window size between 1 and 8 words to create a dictionary of word embeddings. A neighborhood of between 1 and 3 words in the continuous space was considered to reduce the vocabulary (parameter n of Algorithm 1). Finally, the summaries produced by each system have up to 100 words.

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

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

Language	Compression ratio		
	$n = 1$	$n = 2$	$n = 3$
English	11.7%	20.1%	25.3%
French	7.1%	12.3%	16.1%

[4] Term Frequency - Inverse Sentence Frequency.
[5] Site: `https://code.google.com/archive/p/word2vec/`.

In order to evaluate the quality of the summaries, we use the ROUGE system[6], which is based on the intersection of the n-grams of a candidate summary and the n-grams of a set of reference summaries. More specifically, we used ROUGE-1 (RG-1) and ROUGE-2 (RG-2). These metrics are F-score measures whose values belong to $[0, 1]$, 1 for the best result [6].

We evaluate the quality systems using DSVs, CSVs and our approach, which results in 3 versions for each system. The default version uses the cosine similarity as similarity measure for the base, MMR and Sasi systems with DSVs; the TextRank system calculates the similarity between two sentences based on the content overlap of DSVs. In the "cs" version, all systems use the phrase embedding representation for the sentences as described in [5] and employ the cosine similarity as similarity measure. Finally, the "rv" version (our method) uses a reduced vocabulary and the same metrics as the default version with DSVs. After selecting the best sentences, all system versions create a summary with the original sentences.

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

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

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

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

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

in the continuous space and the reduced vocabulary may be incorrect. The "cs" version mitigates the problem with the small vocabulary because this version only analyzes the words of the text that exist in the continuous space. Thus the "cs" version produces better summaries for almost all systems.

6 Conclusion

We analyzed the summary quality of different systems using Discrete Space Vectors and Continuous Space Vectors. Reducing the text vocabulary with a CSV-based similarity using a big training corpus produced better results than the methods described in [5] for English, but lower for French.

As future work, we will increase the French training corpus to extend the dictionary of word embeddings, and use other methodologies to create continuous space vectors, such as [3] and [9]. Furthermore, new methods have still to be devised to fully exploit CSVs to calculate the sentence relevance.

References

1. Balikas, G., Amini, M.R.: Learning language-independent sentence representations for multi-lingual, multi-document summarization. In: 17ème Conférence Francophone sur l'Apprentissage Automatique (CAp) (2015)
2. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR. pp. 335–336 (1998)
3. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: ICML. pp. 160–167 (2008)
4. Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., Varma, V.: TAC2011 multiling pilot overview. In: TAC (2011)
5. Kågebäck, M., Mogren, O., Tahmasebi, N., Dubhashi, D.: Extractive summarization using continuous vector space models. In: 2nd EACL Workshop on Continuous Vector Space Models and their Compositionality (CVSC). pp. 31–39 (2014)
6. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: ACL workshop on Text Summarization Branches Out (2004)
7. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: EMNLP. pp. 404–411 (2004)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
9. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (October 2014)
10. Phung, V., De Vine, L.: A study on the use of word embeddings and pagerank for vietnamese text summarization. In: 20th Australasian Document Computing Symposium. pp. 7:1–7:8 (2015)
11. Pontes, E.L., Linhares, A.C., Torres-Moreno, J.M.: Sasi: sumarizador automático de documentos baseado no problema do subconjunto independente de vértices. In: XLVI Simpósio Brasileiro de Pesquisa Operacional (2014)
12. Torres-Moreno, J.M.: Automatic Text Summarization. John Wiley & Sons (2014)