The Initial Study of Term Vector Generation Methods for News Summarization

Michal Rott

Institute of Information Technology and Electronics,
Studentská 1402/2, 461 17 Liberec, Czech Republic
michal.rott@tul.cz

Abstract. In this paper, I present initial study of new term vector generation methods. The Random Manhattan Indexing and the Skip-gram model were introduced as novel techniques of term vector generation with interesting features. The purpose of this study is to determine whether the methods are suitable for the Summec: A Summarization Engine for Czech. The Summec already use Heuristic, TF-IDF and Latent Semantic Analysis methods for news article summarization. I test quality of generated vectors on the Summec's evaluation set and compare them with existing summarization methods. The novel summarization methods perform by 2 % worse than the LSA method. The evaluation set contains 50 newspaper articles, each annotated by 15 persons. The ROUGE toolkit is used to compare generated summaries with the human references. The above-mentioned evaluation set and the Summec demo are available online at http://nlp.ite.tul.cz/sumarizace.

Keywords: Latent Semantic Analysis, Random Manhattan Indexing, Skip-gram Model, Vector Space Model, Automatic Summarization

1 Introduction

Two novel methods of term vector generation were introduced in 2014. The first one is the Skip-gram model (SGM). Tomas Mikolov introduced very interesting features of the SGM in his paper [1]. This method is able to model relations between words and use them for further analysis. This model gained a lot of attention and is frequently tested in many NLP tasks; such as word clustering [2]. The second method is the Random Manhattan Indexing (RMI). The RMI takes advantage of random vector generation and this leads to extremely quick vector generation. The authors presented its ability to detect similarity of wikipedia pages [3].

I have already tested these methods in task of newspaper articles clustering [4]. Both methods outperformed the Latent Semantic Analysis (LSA). The LSA is used as basis method of text vectorisation for different NLP tasks; e.g. text indexing [5] or summarization [6,7]. In this paper, I want to test performance of the novel methods in task of single-document summarization and compare them with already implemented methods.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2015, pp. 23–30, 2015.

Linguistic properties of Czech language complicate the use of aforementioned methods. Czech words have many forms and their analysis requires lemmatization. The free word order limits language modelling and evaluation of summarization methods. Use of bi-grams and larger n-grams is questionable for evaluational by n-gram co-occurrence. Therefore, I use uni-grams to evaluate implemented methods. The last issues of Czech is rich vocabulary and there are many words representing the same thing. The vector generation algorithm have to unnecessarily train vectors for semantically similar words. Czech thesaurus [8] can be used to minimize influence of synonyms. A preprocessing module of the Summec solves this issues for Czech language and provides limited support for other Slavonic languages.

2 VSM and Summarization

The Vector Space Model is a well mathematically formed construct. The computation of similarity of the documents is reduced to computation of distance between their vectors. The distance can be computed by several metrics. Manhattan metric and cosine similarity are commonly used to compare vectors in NLP.

The main issue of the use of VSM is generation of vectors. The methods were tested: Latent Semantic Analysis, Random Manhattan Indexing and Skip-gram model. The first method is described in paper [9]. The rest will be described further in this paper.

A news article is composed from sentences and they are describing topics of the article. There are two possible ideas how to perform summarization in VSM:

1. Extract sentences of the main topic.
2. Extract the most important sentences of all topics.

The first idea is useful for single-document summarization. A sentence with the longest vector contains probably the most important terms of the document. Lets call this sentence the main sentence of the document. Sentence vectors with similar directions as the main sentence describe the same topic.

For example, Fig. 1 represents a document with 5 sentences and the objective is to extract the two most important sentences. Sentence s1 is clearly the most important sentence (the one with the longest vector) and the most similar sentence to s1 is s2. Hence, this summarization method will extract sentences s1 and s2.

The second idea is more suitable for multi-document summarization or large documents with more topics. The method extracts sentences with the longest and the most distant vectors between each other. The method alternates sentence vectors after every sentence extraction. This can be done iteratively and is described as follows:

1. Compute vectors of sentences from their term frequencies

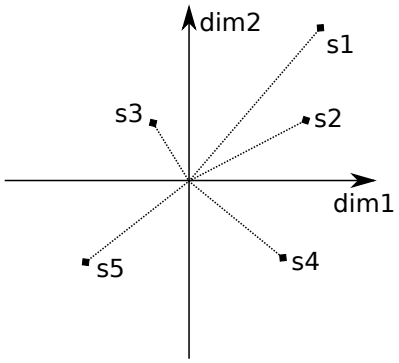

Fig. 1. Vector representation of document sentences in two-dimensional space.

2. Sentence with the longest vector is extracted.
3. The term frequency of words of the sentence are set to zero.
4. Repeat steps 1-3 and stop when extract is complete.

If we look at the Fig. 1, this method will extract sentence s1. After vectors alternation, s5 will be the longest vector. Sentences s1 and s5 will form the extract.

2.1 The Random Manhattan Indexing

The Random Manhattan Indexing method (RMI) was introduced in [3] and the main idea came from Random Projection. The LSA reduces dimensions using low rank approximation of space by Singular Value Decomposition. The main advantage of the RMI is the skip of SVD computation.

The RMI constructs L1 normed vector spaces with reduced dimensionality. It replaces the Euclidean metric with the Manhattan metric. The Manhattan Metric is not sensitive to non-Gaussian noise[1] [10]. Hence, the Manhattan metric yields good results in tasks of text similarity comparison.

The Manhattan metric is described as:

$$d(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=0}^{N} |a_i - b_i| \tag{1}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two vectors from a generated VSM and N is the number of dimensions.

The RMI is a two-step procedure. At first, index vectors are generated for terms of the text. Index vector $\boldsymbol{t}$ is randomly generated with the following probability distribution:

$$t_i = \begin{cases} \frac{-1}{U_1} & \text{with probability } \frac{s}{2} \\ 0 & \text{with probability } 1 - s \\ \frac{1}{U_2} & \text{with probability } \frac{s}{2} \end{cases} \tag{2}$$

[1] Non-Gaussian noise represents peaks in word frequencies. This phenomenon appears when a word is frequently repeated.

where U_1 and U_2 are two independent uniform random variables in (0,1) and s determines the sparseness of the index vectors. Value t_i represents i'th value of the index vector.

The sentence vectors are computed as the sum of index vectors. This sum is described as follows:

$$s_k = \sum_{t \in s} t_k \tag{3}$$

The k'th dimension of sentence vector s is computed as the sum of k'th dimension of every index vector of sentence s.

The problem with OOV words[2] is solved very simply; when a vector for a previously unobserved term is needed, a new vector is generated by probabilistic distribution (2).

2.2 The Skip-gram Model

The training of this model is very efficient and is based on log-linear neural network architecture. The training objective of the Skip-Gram Model is to find the best vector representation of terms in VSM that best predicts the surrounding words in the document.

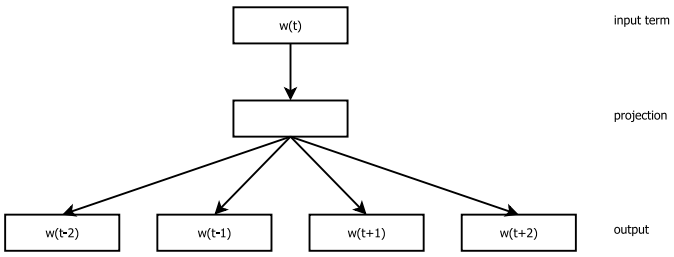

Fig. 2. The Skip-gram model architecture.

Formally, the objective is to maximize the average log probability for a given sequence of terms $t_1, t_2, ...t_N$

$$\frac{1}{N} \sum_{n \in N} \sum_{-c \leq j \leq c, j \neq 0} \log(p(t_{n+j}|t_n)) \tag{4}$$

where c is the size of the term context.

Computing log probability is not efficient because the cost of computing is dependent on the size of the training set. Therefore, the authors of the model define Negative sampling as objective which replaces the log probability computation with logistic regression [1].

[2] Out-of-Vocabulary words are words not observed in training data.

The Subsampling of Frequent Words is used to balance the occurrence of frequent terms (e.g., “být", “a", “i" and “v") with rarer terms that have more of an informational value. The terms’ probability is computed with the formula

$$P(t_i) = 1 - \sqrt{\frac{h}{f(t_i)}} \tag{5}$$

where $f(t_i)$ is the frequency of term t_i in the training data and h is the heuristically chosen threshold. The recommend value is around 10^{-5}.

The equation (3) is used to compute sentence vectors and cosine similarity (6) is used for vector comparison. The main disadvantage is handling of OOV words. The whole model supplemented by new data has to be recomputed to gain vectors for unobserved words. Nevertheless, the SGM produces very interesting spaces with semantically distributed word vectors as shown in [1].

$$d\left(\boldsymbol{a}, \boldsymbol{b}\right) = \frac{\sum_{i=1}^{k} a_i \times b_i}{\|\boldsymbol{a}\| \times \|\boldsymbol{b}\|} \tag{6}$$

3 Experimental Evaluation

3.1 Data for Evaluation

There were no publicly available reference data for evaluation of Czech automatic summarization. Therefore, I created my own test set. This test set contains 50 newspaper articles gathered from Czech news servers. 15 people were asked to produce informative extracts of each article. The articles contained 92089 words in total and were selected from columns on local and international news, economics and culture. The reference extracts contain an average of six sentences. The evaluation set is available on the Summec web page.

3.2 Tools and Metrics Used

The ROUGE [11] was used for evaluation. This toolkit supports various metrics of summarization evaluation. I chose ROUGE-1 metric due to free word order of Czech language. The ROUGE-1 computes co-occurrence of unigrams in the reference and generated summaries. The results obtained using this metric are presented in terms of Recall, Precision and F-score:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad F - score = \frac{2RP}{R + P} \tag{7}$$

where TP, FP and FN are explained in Table 1.

Table 1. The meaning of variables in equation (7) for ROUGE-1.

# unigrams	selected by anotators	not selected by annotators
selected by the system	TP	FN
not selected by the system	FP	TN

3.3 Experimental Setup

All summarization methods use preprocessing module of the Summec. This module offers sentence separation, words lemmatization, stop list, inverse document frequency dictionary and synonyms substitution. The separation of sentences is done using sequence of regular expressions that follows Czech language grammar. The Morphodita [12] is used to lemmatize the input texts. The stop list and IDF dictionary are created using 2.2M newspaper articles. The IDF dictionary contains 491k Czech lemmas. The resulting stop list contains over 200 Czech terms, including the most frequent Czech words and Czech prepositions, conjunctions and particles. Synonyms dictionary 7443 different groups of synonyms with a total of 22856 lemmas.

3.4 Comparison of Summarization Methods

The best performing method of the Summec (TFxIDF) and the LSA method are compared in 3.4 with novel methods. Both novel methods generated the exactly same extracts for our evaluation data. Hence, the result are identical.

Table 2. Comparison of ROUGE-1 score of summarization methods.

method	Recall [%]	Precision [%]	F-score [%]
LSA	55.4	55.1	55.2
RMI	50.7	56.7	53.3
SGM	50.7	56.7	53.3
TFxIDF	62.6	53.3	57.3

The novel methods scored by 1.9 % worse than the LSA and by 4 % worse than the TFxIDF. The result of the RMI methods was anticipated because this method uses random distribution to generate term vectors. By principle, these vectors can not semantically represent document sentences in VSM. In contrast, term vectors generated by the SGM method are represented semantically in space. Therefore, the same result as the RMI is surprising.

4 Conslusion

In this paper, I presented comparison of two news article summarization methods with methods implemented in the Summec. I evaluated their performance on an evaluation set containing 50 Czech news articles. The LSA-based method performs by 1.9 % better then the RMI and SGM methods. Nevertheless, the RMI offers very efficient way how to handle Out-of-Vocabulary words. The SGM has a feature of semantic representation of term vectors in VSM. I want to utilize this two methods in our next work. The demo of the Summec and the evaluation set is available on web page `http://nlp.ite.tul.cz/sumarizace` for public use.

Acknowledgments This paper was supported by the Technology Agency of the Czech Republic (project no. TA04010199) and by the Student Grant Scheme 2015 (SGS) at the Technical University of Liberec.

References

1. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations ofwords and phrases and their compositionality. In: Advances in Neural Information Processing Systems. (2013)
2. Lu, Y., Ji, D., Yao, X., Wei, X., Liang, X.: Chemdner system with mixed conditional random fields and multi-scale word clustering. Journal of Cheminformatics **7** (2015) cited By 3.
3. Zadeh, B.Q., Handschuh, S.: Random manhattan indexing. In: Proceedings - International Workshop on Database and Expert Systems Applications, DEXA. (2014) 203–208
4. M., R., Červa P.: Comparison of term vector generation methods for news clustering. In: 7th Language & Technology Conference, Poznan (Poland). (2015)
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE **41**(6) (1990) 391–407
6. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (2001)
7. Steinberger, J., Ježek, K.: Text summarization and singular value decomposition. In: Proceedings of the Third international conference on Advances in Information Systems. ADVIS'04, Berlin, Heidelberg, Springer-Verlag (2004) 245–254
8. Pala, K., Všianský, J.: Slovník českých synonym (Dictionary of Czech Synonyms, SČS). 3 edn. Lidove Noviny Publishers, Praha (2000)
9. Rott, M., Červa, P.: Summec: A summarization engine for czech. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **8082 LNAI** (2013) 527–535 cited By 0.
10. Weeds, J., Dowdall, J., Schneider, G., Keller, B., Weir, D.: Using distributional similarity to organise biomedical terminology. Terminology **11**(1) (2005) 107–141
11. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings ACL workshop on Text Summarization Branches Out. (2004)

12. Straková, J., Straka, M., Hajič, J.: Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, Association for Computational Linguistics (June 2014) 13–18