SummEC: A Summarization Engine for Czech

Michal Rott and Petr Červa

Institute of Information Technology and Electronics, Technical University of Liberec
Studentska 2, 461 17, Liberec, Czech Republic
{michal.rott,petr.cerva}@tul.cz
https://www.ite.tul.cz/itee/

Abstract. This paper describes a summarization engine developed primarily for the Czech language. Therefore, the engine takes advantage of language-dependent preprocessing modules performing segmentation of the input document into sentences, lemmatization and substitution of synonyms. Our system is also implemented as a dynamic library which can be employed in either a web or a desktop application, and supports a variety of summarization methods. To evaluate the performance of the system, several experiments are conducted in this paper using a set of manually created summaries. The obtained results show that our engine yields an outcome for Czech which is better or at least comparable to other online summarization systems. The above-mentioned reference summaries and the presented summarization engine are available online at http://summec.ite.tul.cz.

Keywords: automatic summarization, single-document summarization, latent semantic analysis, term frequency, inverse document frequency.

1 Introduction

Automatic summarization [1] is a wide scientific discipline which makes use of linguistic, mathematical and computer science knowledge and skills. The primary goal of the discipline is to instruct computers how to distill the most important information from a source (single-document summary) or set of sources (multi-document summary). It is also possible to distinguish between summaries which are created as extracts and those which represent abstracts and contain new sentences or phrases generated by the automatic summarizer [2]. The next traditional distinction is between informative summaries, which can be used instead of the source, and indicative summaries, which allow selection of documents for further, more detailed analysis.

All aforementioned types of summaries can be useful in a wide range of applications. For example, informative summaries of lectures allow the review of course materials [3]. On the other hand, indicative summaries can simulate the work of an intelligence analyst or enable the analyst to more accurately find the most relevant documents. It would also be interesting to have a system providing a multi-document information summary of a story from various newspapers or even directly from TV channels.

It is therefore not surprising that systems for automatic summarization have attracted a lot of attention recently and several evaluations [4] of these systems have been conducted within the Text Analysis Conferences (TAC)[1]. Several summarization

[1] http://www.nist.gov/tac

I. Habernal and V. Matousek (Eds.): TSD 2013, LNAI 8082, pp. 527–535, 2013.

systems also exist that are available online, such as the Open Text Summarizer[2], Free Summarizer[3] or Text Compactor[4].

Unfortunately, all of these systems, as well as others, are language-dependent and their use for the highly inflective Czech language is limited (as proven by results presented in Section 4.6) Note that only the Open Text Summarizer provides direct support for the Czech language. There is also the Almus[5] summarizer developed in the Czech republic, but the data files released on Almus web pages only allow the creation of summaries from English texts.

Therefore, we decided to develop a new Summarization Engine for Czech (SummEC). The motivation for this development also stems from the fact that we cooperate with a media-monitoring company which operates in the Czech, Slovak and Polish market and delivers news digests particularly from the print sources. These digests could be created automatically by a summarization engine.

Hence, the current version of our system produces informative summaries, which are created on a single-document basis. The multi-document extension and support for other (Slavic) languages is under development.

Note that our summaries are created as extracts and with the use of a text-preprocessing module, which performs language-dependent operations such as substitution of synonyms and lemmatization, that are important for the resulting accuracy of the system (as proven in Section 4.5).

The rest of the paper is structured as follows: The next section describes the architecture and modules of the SummEC system. Section 3 then reviews the principles of summarization methods that are supported by SummEC. Experimental evaluation of these methods is then given in Section 4, where we also compare our results with those obtained using several online summarization tools. The last section 5 then concludes this paper.

2 Description of the SummEC System

2.1 System Architecture

The overall scheme of SummEC is depicted in Figure 1. The figure shows that the system is composed of a preprocessing and summarization modules and that the output from the former module serves as the input to the latter one.

The figure also demonstrates that SummEC is implemented as a dynamic library which can be employed in either a web or a desktop application. The output summary (i.e., the sequence of the most important sentences) can be converted to the xml format for evaluation and further processing, or to the format appropriate for display on a web page.

The preprocessing module is detailed in the following subsection. The summarization module provides support for several summarization methods. Their principles are described in Section 3.

[2] http://libots.sourceforge.net

[3] http://freesummarizer.com

[4] http://textcompactor.com

[5] http://textmining.zcu.cz/?lang=en&section=download

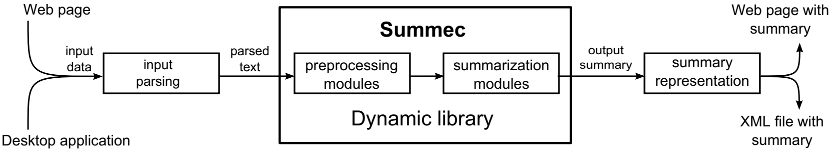

Fig. 1. The scheme of the developed Summarization Engine for Czech (SummEC)

2.2 The Preprocessing Module

The preprocessing module converts the input text to its normalized form. This process is carried out in three consecutive phases as follows:

Sentence Segmentation: The text is split into sentences in the first step. The splitting routine takes into account Czech words ending with the symbol '.' such as academic titles (Ing.), military titles (gen.) or abbreviations.

Lemmatization: In the next step, every sentence is lemmatized using an external morphological tool Fmorph[6]. This analyzer was chosen as it has a higher latency than newer Czech taggers Morče[7] and Compost[8]. This factor is important, because SummEC also performs summarization in the online mode (over the web interface).

Note that, after lemmatization, the resulting text does not just contain lemmas, but also word forms, such as numbers or typing errors, which cannot be lemmatized. We further call all of these items of the lemmatized text as terms.

Substitution of Synonyms: The goal of the third step is to substitute all synonyms of every lemma using one preferred form. The substitution is based on the use of a lemmatized dictionary of synonyms, which contains 7443 different groups of synonyms with a total of 22856 lemmas. These items are compiled from two sources. The first is the Czech version of the project Wiktionary[9]. The second is the Thesaurus project[10].

3 Supported Summarization Methods

3.1 The Heuristic Method

The heuristic method [1] is based on a natural idea that a word which occurs frequently in the input document is important and should therefore be presented in the resulting

[6] http://ufal.mff.cuni.cz/pdt/Morphology_and_Tagging/Morphology/index.html

[7] http://ufal.mff.cuni.cz/morce/

[8] http://ufal.mff.cuni.cz/compost/

[9] http://cs.wiktionary.org

[10] http://packages.debian.org/sid/myspell-cs

summary of the document. However, common words exist for every language, e.g., prepositions or conjunctions, which are generally considered irrelevant to the meaning or topic of the given document. These words should be included in a stop list.

An advantage of the heuristic method is that it is simple to implement, because it works in three simple steps, as follows:

At first, the frequency of occurrence, i.e., the term frequency (TF), is calculated for every term from the input document, excluding those in the stop list. In the second step, the score of each sentence is given as a sum over TFs of all words in the given sentence. Finally, the sentences with the highest scores are included in the resulting summary.

3.2 The TF-IDF Method

The TF-IDF method [5,6] represents a modification of the heuristic method. It does not rely on a stop list; instead, it assigns a weight to the frequency of each term in the sentence by its inverse document frequency (IDF). For this purpose, it is necessary to first create an IDF dictionary containing IDF values for all terms t from the set of training documents D.

These IDF values can be expressed as:

$$IDF(t) = log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{1}$$

where $|D|$ is the total number of training documents and $|\{d \in D : t \in d\}|$ is the number of documents containing the term t.

Note that the previous equation demonstrates that the resulting IDF value is low for common terms. This is why the IDF method does not require a stop list.

The score of each sentence s from the input document is then simply given as:

$$score(s) = \sum_{t \in s} TF(t) \times IDF(t) \tag{2}$$

Similar to the heuristic method, the output summary is created from sentences with the best scores. However, the resulting summary then usually contains very similar sentences, which are composed of the same terms with a high value of the product $TF(t) \times IDF(t)$. To eliminate these redundant sentences, an enhanced $TF(t) \times IDF(t)$ approach was proposed in [7]. This method is carried out in the following steps:

1. The TF values are determined for all terms in the input document.
2. The TF x IDF score is calculated for each sentence.
3. The sentence with the highest score is added to the output summary.
4. The TF values for all terms from this sentence are set to zero.
5. Steps 2-4 are repeated until the summary has the required number of sentences.

3.3 Latent Semantic Analysis

The Latent Semantic Analysis (LSA) [8] is inspired by latent semantic indexing. Therefore, it employs the singular value decomposition (SVD) to the matrix $\boldsymbol{A}$, in which each column vector represents the weighted TF vector of one sentence of the input document. That means that, when the document contains m terms and n sentences, the matrix $\boldsymbol{A}$ has a size of $m \times n$. The SVD of $\boldsymbol{A}$ is then defined as:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3}$$

where $\boldsymbol{U}$ is an $m \times n$ column-orthonormal matrix of left singular vectors, $\boldsymbol{\Sigma} = diag(\sigma_1, \sigma_2, ..., \sigma_n)$ is an $n \times n$ diagonal matrix whose diagonal elements represent non-negative singular values sorted in descending order, and $\boldsymbol{V}$ is an $n \times n$ orthonormal matrix of right singular vectors.

The authors in [8] show that the matrix $\boldsymbol{V}$ describes an importance degree for each topic of the document in each sentence. Hence, the resulting summary is created by choosing the most important sentences as follows:

1. The summary does not contain any sentence, $k = 1$.
2. The matrix $\boldsymbol{A}$ is constructed and the SVD of this matrix is performed. Each sentence of the document is then represented in the singular vector space by the column vector of $\boldsymbol{V}^T$.
3. The k'th vector from the matrix $\boldsymbol{V}^T$ is selected.
4. The sentence that has the largest index value with this vector is included in the summary
5. Similar to the TF x IDF method, the value of k is incremented and steps 3 and 4 are repeated until the summary has the required number of sentences.

As mentioned above, each column vector of the matrix $\boldsymbol{A}$ represents the weighted TF vector of one sentence. The weighting can be performed using several different methods [9]. The recent version of SummEC supports two basic approaches. The first takes advantage of the IDF values defined above. In the second approach, the frequency of each term in the sentence is normalized by its frequency over the whole document.

3.4 Modified LSA

The approach described above was modified in [10] using elements of the diagonal matrix $\boldsymbol{\Sigma}$. The score of each sentence s is then expressed as:

$$score(s) = \sqrt{\sum_{i=1}^{p} v_{k,i}^2 \cdot \sigma_i^2} \tag{4}$$

where p is the number of chosen dimensions of the new space, $v_{k,i}$ is the i'th element of the k'th column vector of the matrix $\boldsymbol{V}$, and σ_i is the corresponding element of the matrix $\boldsymbol{\Sigma}$. The authors in [10] also suggest choosing only the dimensions whose singular values were smaller than half of the highest singular value.

4 Experimental Evaluation

4.1 Data for Evaluation

Unfortunately, no publicly available reference data exist for evaluation of automatic summarization in the Czech language. Therefore, we had to create our own test set:

we asked 15 persons to produce informative extracts of 50 different newspaper articles. The articles contained 92089 words in total and were selected from columns on local and international news, economics and culture. The resulting extracts contain an average of six sentences from each article (i.e., 20 % of the sentences from each article) and are made public on the SummEC web pages.

4.2 Tools and Metrics Used

We used the toolkit ROUGE [11] for evaluation, which supports various metrics to compare automatically-produced summaries against manually-produced references. In this paper, we chose the metrics ROUGE-1 and ROUGE-W. The former is based on co-occurrence of unigrams. The latter represents statistics based on Weighted Longest Common Subsequence (WLCS) [11]. The weight was set on the value of 1.2 in our case.

In following subsections, the results obtained using these metrics are presented in terms of Recall, Precision and F-score. These are defined for ROUGE-1 as:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad F - score = \frac{2RP}{R + P} \tag{5}$$

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

Table 1. The meaning of variables in equation (5) for ROUGE-1

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

A more complex definition of Precision, Recall and F-score for ROUGE-W can be found in [11].

4.3 Experimental Setup

The summarization methods reviewed in Section 3 make use of a stop list and an IDF dictionary. In this work, both of these components are created using 2.2M newspaper articles. The resulting stop list contains 283 items, including the most frequent Czech words and Czech prepositions, conjunctions and particles. The IDF dictionary has 491k items. They represent all terms from the lemmatized articles whose frequency of occurrence was higher than five.

4.4 Comparison of Supported Summarization Methods

The aim of the first experiment performed was to compare the results of individual summarization methods that are supported by SummEC. The experiment was carried out using all preprocessing modules and the stop list. The obtained results are presented in Table 2. In this table, the method denoted as LSA-IDF corresponds to the enhanced version of LSA with weighting based on IDF. In contrast, LSA-TF stands for the enhanced LSA using TF normalization.

It also should be noted that in this experiment, we take advantage of the approach based on increasing the frequency of terms that are important for the topic of the document. We suppose that these topic terms are those included in the title of each document and we multiply their TF values by two.

The presented results show that the TFxIDF method yielded the highest F-score for the metric ROUGE-1 (57.3 %) as well as ROUGE-W (30.0 %). This approach also led to the best Recalls of 62.6 % and 35.5 % respectively. In contrast, the highest Precisions of 55.2 % and 28.9 % were reached by LSA-IDF. It is also evident that the worst F-scores were obtained by using LSA-TF.

Table 2. Comparison of results of individual summarization methods supported by SummEC

method	ROUGE-1			ROUGE-W		
	Recall [%]	Prec. [%]	F-score [%]	Recall [%]	Prec. [%]	F-score [%]
Heuristic	57.2	54.3	55.3	30.3	27.9	28.3
TFxIDF	62.6	53.3	57.3	35.5	26.6	30.0
LSA-IDF	55.4	55.2	55.1	28.6	28.9	28.4
LSA-TF	57.6	50.1	53.3	28.6	22.2	24.6

4.5 Performance of SummEC's Components

In the second series of experiments, individual components of our system were activated gradually to show their contribution to the system's overall accuracy. We employed the TFxIDF method, which yielded the best results in the previous experiment. The obtained results are presented in Table 3.

This shows that the highest absolute increase in F-score was reached for both metrics by using lemmatization. The other components and approaches yielded only a small additional improvement of this measure. The exception is the stop list, which increased the F-score of ROUGE-W from 29.4 % to 30.3 %.

4.6 Comparison with other Online Systems

The final experiment (see Table 4) compares the results yielded by SummEC (using TFxIDF) with several online summarization systems. We can see that SummEC outperformed not only the systems without explicit support for Czech, as we expected, but the OTS system as well. This proves that SummEC is a useful tool for automatic summarization of documents in the Czech language.

Note that the worst results reached by the Free Summarizer are caused by the fact that this tool does not correctly accept the Czech set of characters. For that reason, some other online systems were not evaluated at all.

Table 3. Contribution of individual components to the overall performance of our engine

component	ROUGE-1			ROUGE-W		
	Recall [%]	Prec. [%]	F-score [%]	Recall [%]	Prec. [%]	F-score [%]
no component	58.3	50.6	53.9	31.0	23.8	26.6
lemmatization	63.9	51.6	56.9	37.1	24.9	29.5
+ stop list	63.0	52.5	57.0	35.7	25.5	29.4
+ topic terms	61.6	53.7	57.1	35.2	27.3	30.3
+ synonyms	62.6	53.3	57.3	35.5	26.6	30.0

Table 4. Comparison of results yielded by SummEC and several online summarizers

system	ROUGE-1			ROUGE-W		
	Recall [%]	Prec. [%]	F-score [%]	Recall [%]	Prec. [%]	F-score [%]
SummEC	62.6	53.3	57.3	35.5	26.6	30.0
Text Compactor	56.1	51.6	53.2	32.0	27.6	28.5
Free Summarizer	25.9	36.0	29.3	7.4	14.2	9.0
Open Text Summ.	50.8	54.7	52.4	27.1	31.8	28.8

5 Conclusion

In this paper, we presented our summarization engine developed for the Czech language and evaluated its performance on the set of manually created reference summaries. This evaluation demonstrated that a) the TFxIDF method is capable of producing the best automatic summaries and b) the lemmatization module is an important component of the system, because Czech is a highly inflective language. The comparison of SummEC's results with those yielded by several online summarization systems showed that our engine produces summaries of high accuracy. As previously mentioned, the online version of our engine and the reference summaries are available for free at http://summec.ite.tul.cz. Support for other Slavic languages, particularly for Slovak, Polish and Croatian, is under development.

Acknowledgments. This paper was supported by the Technology Agency of the Czech Republic (project no. TA01011204) and by the Student Grant Scheme (SGS) at the Technical University of Liberec.

References

1. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. 2, 159–165 (1958)
2. Jing, H., McKeown, K.R.: Cut and paste based text summarization. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, NAACL 2000, pp. 178–185. Association for Computational Linguistics, Stroudsburg (2000)
3. Fujii, Y., Kitaoka, N., Nakagawa, S., Nakagawa, S.: Automatic extraction of cue phrases for important sentences in lecture speech and automatic lecture speech summarization. In: INTERSPEECH, pp. 2801–2804 (2007)

4. Mani, I., Klein, G., House, D., Hirschman, L., Firmin, T., Sundheim, B.: Summac: a text summarization evaluation. Nat. Lang. Eng. 8, 43–68 (2002)
5. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
6. Skorkovská, L.: Application of lemmatization and summarization methods in topic identification module for large scale language modeling data filtering. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2012. LNCS, vol. 7499, pp. 191–198. Springer, Heidelberg (2012)
7. Vanderwende, L., Suzuki, H., Brockett, C., Nenkova, A.: Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. Inf. Process. Manage. 43, 1606–1618 (2007)
8. 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)
9. Berry, M., Browne, M.: Understanding Search Engines. Society for Industrial and Applied Mathematics, Philadephia (2005)
10. Steinberger, J., Ježek, K.: Text summarization and singular value decomposition. In: Yakhno, T. (ed.) ADVIS 2004. LNCS, vol. 3261, pp. 245–254. Springer, Heidelberg (2004)
11. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings ACL Workshop on Text Summarization Branches Out (2004)