Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CONTRASTIVE SUMMARIZATION: COMPARING OPINIONS OF CZECH SENATORS

[1]MICHAL CAMPR, [2]KAREL JEŽEK
[1,2]Department of Computer Science and Engineering, FAS, University of West Bohemia
E-mail: [1]mcampr@kiv.zcu.cz, [2]jezek_ka@kiv.zcu.cz

ABSTRACT

In this paper, we present a novel approach to contrastive summarization, i.e. a specific type of summarization, which aims to compare two documents (or groups of documents) on semantic and also sentiment level. The final output of contrastive summarization is a pair of summaries, depicting what topics are most often discussed with the largest difference in opinions of the authors. We explore the possibilities of combining the latent semantic information with the information about the opinions of the authors. First, we describe related works, which show, that this problem can be approached from many different directions. Next, we present our own algorithm, based on Latent Semantic Analysis, which computes scores for excerpts of the original text and based on these, it chooses best excerpts that should be included into the final summaries. Finally, we present the evaluation of our algorithm, using speeches from Czech senate.

Keywords: *Contrastive Summarization, Semantic Analysis, Sentiment analysis, LSA, Classification*

1 INTRODUCTION

Opinions appear in almost all aspects of our activities and thus they influence our behavior, our perception of reality and affect how we see and evaluate the world around us. This is the reason why we often seek the opinions of others, especially when we are about to make a decision, whether it is about buying new things or choosing where to go on a trip. Sentiment analysis, the study of opinions, sentiments and emotions, has been the subject of study for a considerable amount of time now and still presents a very challenging task for Natural Language Processing (NLP). It is also widely studied in data mining, web mining and text mining and it has spread from computer science to other areas, such as management or social sciences due to its importance to society as a whole.

With the rapid growth of the internet in the last decade, an increasing number of people express their opinions on the web and particularly on social media, e.g. in the form of reviews, forum discussions or blogs. With so many new opinionated data recorded on so many subjects, there is only a little chance that any individual could keep track of them all and be familiar with the entire contents of these texts. This particular problem, i.e. being able to quickly assess the content of a document, can be at least facilitated by another widely studied area of NLP, called automatic summarization. All summarization algorithms have the same purpose, which is to analyze the content of the given document and then construct a short summary, which should contain the most important information from the original document.

The purpose of this paper is to present our novel approach to a task called contrastive summarization. This task aims to combine the current knowledge from two NLP areas mentioned in the previous paragraphs, i.e. sentiment analysis and automatic summarization. To be more precise, we attempt to combine sentiment analysis with the, so called, comparative summarization. This particular task deals with semantically comparing two documents (or two groups of documents) and producing summaries, depicting the most significant information, in which the documents differ.

The evaluation of similar methods is generally conducted on user reviews of movies or electronics, which often discuss similar topics linked to a particular domain. However, due to the upcoming senate elections (in the time of conducting these experiments), we decided to utilize the political texts from Czech senate and provide a tool for future voters to easily evaluate the opinions of current senators. The speeches from Czech senate also provide additional challenge,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

because they are unstructured and are not linked to a particular domain.

The following section discusses methods, which are closely related to ours and provides basic descriptions of their solutions. Section 3 provides a detailed description of our algorithm and Section 4 contains the evaluation of its performance. Finally, in Section 5, we discuss our findings and briefly outline our future research.

2 RELATED WORK

Several papers dealing with comparative summarization have already been published. The following examples show, that this particular problem can be approached from many different angles and using various methods.

The first example [1] proposes a sentence selection method (based on a multivariate normal generative model) for extracting sentences which represent specific characteristics of multiple document groups. In [2] ,on the other hand, is proposed an approach to generating comparative news summaries. The task is formulated as an optimization problem of selecting proper sentences to maximize the comparativeness within the summary and the representativeness of the summary to both topics. The optimization problem is addressed by using a linear programming model. The last example is [3], where is proposed a framework for multi-document summarization using the minimum dominating set of a sentence graph which is generated from a set of documents.

Similarly, several examples, dealing with contrastive summarization, can be provided. Firstly, [4] approach it as a classification problem at sentence level using a Naive Bayes and regularized logistic regression models. In [5], on the other hand, is proposed a machine-learning method that applies text-categorization techniques to just the subjective portions of the document to determine the sentiment polarity (classifying a movie review as 'thumbs up' or 'thumbs down'). Extracting these portions can be implemented using efficient techniques for finding minimum cuts in graphs.

We have also presented several papers regarding comparative summarization, e.g. [6], [7] and now we intend to extend the functionality of our algorithm, which deals with combining semantic topics of documents, with the current knowledge in the area of sentiment analysis. The final algorithm could be used, for example, to compare two different product reviews. If a user is trying to decide which product (e.g. a computer or a television) to buy, he could use our summarizer to compare different reviews to see, which attributes of the particular product are the most discussed and what opinions do the authors hold about them. Another example can be in politics. We can use our summarizer to compare the speeches of two different political speakers about the same topic (e.g. a new law proposal) and compare them. The final summary can tell us in which points the speakers agree or disagree. We chose these examples on purpose to outline our current and our future goals.

We intend to apply our algorithm on a natural text taken from official sources, such as the records from the Czech senate (as discussed further in the paper), instead of texts from social media or product reviews from e-shops and forums. Although the analysis of these texts is more commercially attractive, they are very often structured and written using colloquial language. These features are inconsistent with our idea of extractive summarization, and we intend to focus on them in the future and thus we currently utilize official texts written using literary language.

3 ALGORITHM

This chapter will, in detail, describe our novel algorithm. It ultimately consists of three steps (see Figure 1), each described in its own section:

- Input text preprocessing (section 3.1)
- Semantic processing (section 3.2)
- Summary creation (section 3.3)

3.1 Input text preprocessing

The purpose of this step is to transform the two input groups of documents D_1 and D_2 into two sets of sentences, containing all the information needed for further computations (e.g. word frequencies and their sentiment values). Since we are dealing with analysis of natural texts, several preprocessing steps are needed before we start to examine the semantic information contained within it. The first step is splitting the input text into sentences, which we solved using the *nltk* [8] library in python. Similarly, we performed word tokenization to extract individual words. Each token, that contains only numbers or is a stop-word, is ignored. Other tokens are lemmatized, and also, their sentiment values are decided. Note, that this is the only language-dependent step. The main reason is, that it includes lemmatization and sentiment analysis algorithms based on language-dependent lexicons.

ISSN: -3195

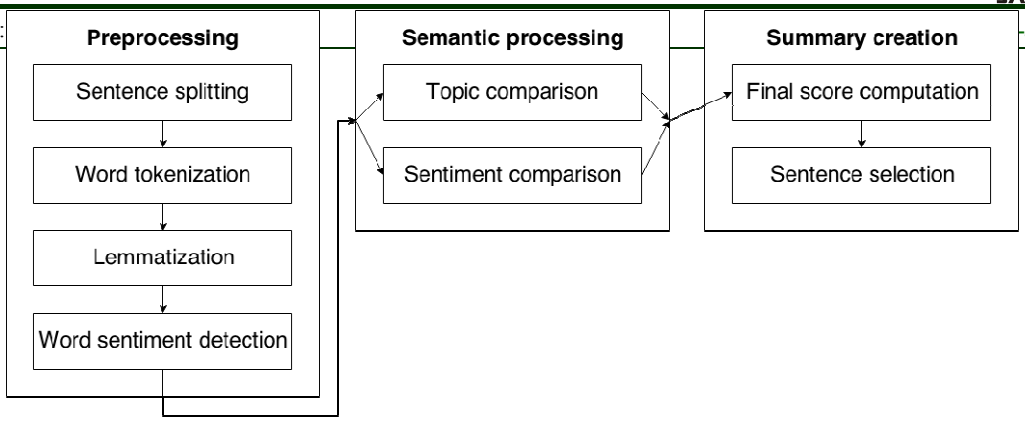

Figure 1: The Structure Of Our System.

3.2 Semantic processing

In this step, we utilize the output from the previous one and perform semantic analysis of the extracted sentences and their words. We perform this analysis on two independent levels, that is topic comparison and sentiment comparison.

3.2.1 Topic comparison

Because our goal is to select pairs of sentences (from two documents), which depict the differences of opinions about a given topic, we need to be able to compare sentences by topics discussed in them. In order to do that, we have to utilize some latent topic model. In our previous papers, we experimented with Latent Dirichlet Allocation (LDA) [9] and Latent Semantic Analysis (LSA) [10]. The problem of joint sentiment-topic modeling for opinion summarization, based on LDA, has been explored in several papers [11], [12], [13] and [14]. However, our previous experience with LSA, following the works of [15], [16] and [17], show, that it provides better results than LDA.

As was discussed in the previous step (Section 3.2), we created two sets of sentences from D_1 and D_2, containing n_1 and n_2 elements respectively. The next step is creating a $m \times n$ matrix A, where $n = n_1 + n_2$ and m is the total number of terms that appear in all sentences in both documents. Each column of A represents sentence s and each row represents term t. In order to separate the processing of sentiment from semantic information, we include here only terms, which are not associated with any sentiment value (see section 3.2.2). There are several methods on how to compute the elements of matrix A representing term frequencies in sentences, and among the most common are:

- Boolean model - equals to *1* if t occurs in s and *0* otherwise
- Term frequency - raw number of times that term t is in the sentence, depicted as *tf(t, s)*
- Logarithmically scaled term frequency *log(tf(t, s) + 1)*
- Augmented frequency - to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document
- Tf-Idf
- Entropy

In our experiments, we chose to explore the last two methods, i.e. Tf-Idf and Entropy. The Tf-Idf is a product of term frequency *tf(t, s)* (as mentioned above) and inverse document frequency:

$$idf(t,S) = \log \frac{N}{|\, s \in S : t \in s \,|}, \qquad (1)$$

where S is the sentence set, N is the size of set S and $|s \in S : t \in s|$ is the number of sentences from S where the term t appears. The final value is then computed as:

$$tfidf(t,s,S) = tf(t,s) \cdot idf(t,S). \qquad (2)$$

Matrix A constructed using term entropy contains elements computed as:

$$a_{ts} = L(t,s) \cdot G(t,S), \qquad (3)$$

where *L(t, s)* is a boolean value (*1* if term t is present in sentence s, *0* otherwise) and *G(t, s)* is the global weight of term t in the whole document set:

$$G(t,S) = 1 - \sum_j \frac{p_{ts} \cdot \log(p_{ts})}{\log(n)}, p_{ts} = \frac{tf(t,s)}{g_t}, \qquad (4)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

where g_t is the total number of times that term t occurs in the whole document set and n is the number of sentences.

We include sentences from both documents, so that we are able to project them into the same semantic space, ensuring the best possible comparison. With matrix A finished, we apply the Singular Value Decomposition (SVD) on it.

The SVD of A is defined as $A = U\Sigma V^T$, where $U = [u_{ij}]$ is an $m \times n$ matrix and its column vectors are called left singular vectors. Σ is a square diagonal $n \times n$ matrix and contains the singular values. $V^T = [v_{ij}]$ is an $n \times n$ matrix and its columns are called right singular vectors. This decomposition provides latent semantic structure of the input documents represented by A. This means, that it provides a decomposition of documents into n linearly independent vectors, which represent the main topics contained in the documents. If a specific combination of terms is often present within the document set, it is represented by one of the singular vectors. And furthermore, the singular values contained in the matrix Σ represent the significance of these topics. An important fact for us is, that matrix U contains orthogonal vectors, which form a base of a new semantic space. The values contained in each singular vector can be considered to be weights of terms for each topic. This can be used to perform a projection of the original vectors from matrix A into the new semantic space (based on topics rather than terms) by multiplying A^T with $U * \Sigma$ (we multiply by Σ to include also the importance of topics), marked as F.

The columns of matrix F now contain vectors, which represent the original sentences by the means of latent topics, which we obtained using SVD. Since the original matrix A contained sentences from both document sets D_1 and D_2, we split the matrix F appropriately into matrices F_1 and F_2. We can now use vectors from these matrices to compare the sentences of both document sets. We do this by computing the cosine similarity between vectors of F_1 and F_2:

$$sim(t) = \frac{\sum_{t=1}^{m} F_1[t,i] * F_2[t,j]}{\sqrt{\sum_{t=1}^{m} F_1[t,i]^2} * \sqrt{\sum_{t=1}^{m} F_2[t,j]^2}}, (5)$$

where i is the index of the sentence from D_1 and j is the index of sentence from D_2. We store the similarity values $sim(t)$ in a $n_1 \times n_2$ matrix SIM_t, i.e. we compare each sentence from set D_1 with each sentence from set D_2.

3.2.2 Sentiment comparison

Before comparing the sentiment values of sentences, we need to compute them in the first place. As was mentioned in the section 3.1, a lexicon-based algorithm is used to determine, if a word is considered to be opinionated. Each word is assigned a value depending on its positivity or negativity. We also take into account negation words, such as 'no' or 'not' (or language specific equivalents), so that phrases, such as 'not good', are processed accordingly. Each sentence is assigned a sentiment value based on the sentiment values of words it contains. We explored several options of how to compute the sentiment value of a sentence and the results are in the Section 4. The main two problems to consider are the following:

1. How to treat opinionated words with different strength of sentiment.
   - *No sentiment* - sentiment is not included at all. The purpose of this option is to provide a baseline for experimentation with different values.
   - *No differentiation* - each sentiment word has its sentiment value set to *1* if positive, *-1* if negative, *0* otherwise. This option is depicted as '11' in the results (Section 4. 2).
   - *Sentiment differentiation* - the strength of the sentiment of words is reflected in the final score: words with strong sentiment have the value *2* or *-2*, *1* or *-1* for normal sentiment, *0* otherwise. This option is depicted as '12' in the results.
2. How to compute the final sentiment score of a sentence.
   - Static - each word in a sentence contributes with its whole sentiment value.
   - Fraction - sentiment value of a word is divided by the number of already processed sentiment words in the given sentence. Using this option practically means, that we put the highest weight on the first opinionated word.

After computing the sentiment value, we can compare sentences against each other. We compute the difference between these sentiment values for each pair of sentence and store these values in a $n_1 \times n_2$ matrix $DIFF_s$.

3.3 Final score and sentence selection

We have now computed the topic similarity as well as the sentiment similarity of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

sentences. So we create a final matrix F, which aggregates these two scores. We experimented with two formulas (see Section 4 for results):

$$F(i,j) = SIM_t(i,j) + (SIM_t(i,j) * DIFF_s(i,j)), \quad (6)$$

$$F(i,j) = p * SIM_t(i,j) + (1-p) * (SIM_t(i,j) * DIFF_s(i,j)), \quad (7)$$

where i is the index of a sentence from D_1 and j is the index of a sentence from D_2. p is a parameter (value between 0 and 1) to change the ratio between topic similarity and sentiment difference.

Both formulas are basically the sums of two parts. The first part depicts the topic similarity between two sentences, and the second one adds the sentiment difference into the final score. The main problem here is, that each matrix contains values of different orders - SIM_t contains real numbers between -1 and 1, but $DIFF_s$ contains real numbers from 0 to infinity. This is the reason, why we used the multiplication $DIFF_s(i, j)*SIM_t(i, j)$ to bring these values to the same order of magnitude. We experimented with other formulas, however the two presented here provided the best results.

With everything computed, we can now search for maximum values of F which marks two sentences on *i-th* and *j-th* coordinate with the best topic similarity and sentiment difference. We also set the according row and column to zero, so that the same sentence is not selected twice.

4 EVALUATION

The greatest issue with evaluating the problem of contrastive summarization is the lack of publicly available dataset, which would be annotated specifically for this task. Because of this, we decided to evaluate our task alternatively using an extrinsic method [18], specifically classification. The main idea for this type of evaluation is to be able to classify the summarized document the same way and into the same class as the original (not summarized) full document. This way, we can assume, that the summary is a good representation of the original document.

4.1 Classification

We created our dataset by downloading speeches given by senators in the Czech senate between March 2008 and November 2013. Each senator was labeled with his political affiliation and this label was used for training and testing the classifier. It means, that we attempted to assign a political party depending on the content of the speech. Moreover, we filtered the speeches, so that for each political party there are at least 500 speeches. Also, we decided to use only speeches, which are longer than 100 characters. This filtering results into 13,135 speeches from three major political parties for classification:

- ODS - 5,638 speeches
- CSSD - 6,212 speeches
- KDU-ČSL - 1,285 speeches

Each speech (considered as one document) was preprocessed similarly as described in Section 3.1, i.e. tokenized into words and each word was lemmatized. We then computed Tf-Idf weights for these words and used them as features for classification. We also experimented with different features, such as topic weights/probabilities provided by LSA or LDA respectively. However, using these features generally proves to perform worse (concerning classification tasks) than simpler, frequency based, features.

We tested seven commonly used classifiers to classify the senate speeches. We run a 10-fold cross-validation using each classifier (trained on full speeches) and their *Precision*, *Recall* and *F1* scores are available in Table 4. Apparently, the best scoring classifier is the Support Vector Machine trained with Stochastic Gradient Descent, with *F1* score of *0.8326*.

Table 1: Precision, Recall and F1 scores of different models for classification of senate speeches.

	P	R	F1
Linear SVM	0.7502	0.7154	0.7044
SGD trained SVM	**0.8399**	**0.8367**	**0.8326**
Gaussian Naive Bayes	0.7305	0.7000	0.6866
Multinomial Naive Bayes	0.7511	0.7310	0.7171
Regularized Logistic Regression	0.7847	0.7737	0.7662
Nearest Neighbors	0.5733	0.6296	0.5999
Random Forest	0.7855	0.7720	0.7615

4.2 Experiments

The chosen classifier (SGD trained SVM) can be used to evaluate the quality of our summaries, i.e. if a summary represents the full document well enough to be classified into the same class as the original.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We conducted a series of experiments, where we explored different ways of computing the sentiment value and topic similarity score, as discussed in Sections 3.1 and 3.3. The final results, using Equation 6, can be seen in Table 2. The first row lists two types of frequencies (see 3.2.1) and the first column shows different types of sentiment scoring (see Section 3.1).

Table 2: Classification agreement [%] of classifying the original document and its summary, using Equation 6.

Sentiment	Entropy	Tf-Idf
none	0.83979	0.92798
11 static	0.85654	0.93507
12 static	0.91901	0.93843
11 fraction	0.88168	0.92107
12 fraction	0.90123	0.96962

We also experimented with changing the ratio between topic similarity of sentences and their sentiment difference, as was discussed in Section 3.3. The parameter p was given values between *0* and *1* with step *0.1*. Figure 2 shows results only for Tf-Idf, since entropy provided overall lower values.

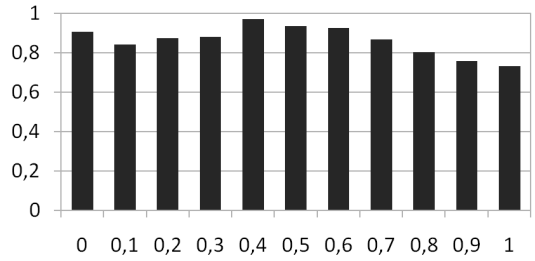

Figure 2: Classification agreement using Tf-Idf and 12-fraction sentiment, with changing the parameter p in Equation 7.

The worst classification agreement score of *71.13%* was provided by using entropy and no sentiment (using Equation 7 with parameter $p = 1$). On the other hand, the best score of *96.97%* is provided by using Tf-Idf with 12-fraction sentiment (using Equation 7 with parameter $p = 0.4$). An interesting fact is, that in most cases, using entropy for building the frequency matrix A performs worse than using Tf-Idf.

The results from all our experiments show, that the classification agreement can significantly change with different settings of aggregating the sentiment values with topic similarity, and thus it is apparent, that any future research in this area can yield some very interesting results.

5 CONCLUSION

The area of contrastive summarization is relatively new and even though there have been several experiments before, it continues to be a very challenging task. Some already published papers dealt with summarizing only specific types of texts, such as structured product reviews. Other papers approached this problem via extracting only the most important words or the, so called, aspects. Our intention was to design an algorithm, which would be able to analyze unstructured texts and extract the most important excerpts from it. Our approach was even more challenging, because we aimed to process texts in Czech, which is a highly inflective language. Also, to our knowledge, no previous work has been published, which deals with this particular problem and in this language.

Though we are still in the process of perfecting our evaluation method, the results show an interesting fact, that including sentiment into the computation improves the quality of the summary.

Because the algorithm, that we used, is the most simple option how to approach sentiment analysis, we intend to utilize more sophisticated machine learning algorithms in the future. We also plan to use other methods for more precise evaluation of our algorithms, namely ROUGE. This, however, requires creating an annotated dataset, which is not yet complete.

6 ACKNOWLEDGEMENT

This project was supported by grant SGS-2013-029 Advanced computing and information systems.

The access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme 'Projects of Large Infrastructure for Research, Development, and Innovations' (LM2010005) is highly appreciated.

REFRENCES:

[1] Wang, Zhu, Li and Gong. 2009 Comparative document summarization via discriminative sentence selection. Proceeding of the 18th ACM conference on Information and knowledge management, pp. 1–18.

[2] Huang, Wan, and Xiao. 2011 Comparative News Summarization Using Linear Programming. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers, 2:648–653.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3] Shen, Ch. and Li, T. 2010 Multi-document summarization via the minimum dominating set. Proceedings of the 23rd International Conference on Computational Linguistics, pp. 984–992.

[4] Beineke, Hastie, Manning and Vaithyanathan. 2003 An exploration of sentiment summarization. Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications.

[5] Pang, B. and Lee, L. 2004, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.

[6] Campr, M. and Ježek, K. 2012, Comparative summarization via Latent Semantic Analysis. Latest Trends in Information Technology, pp. 279–284.

[7] Campr, M. and Ježek, K. 2013, Comparative Summarization via Latent dirichlet Allocation. DATESO, pp. 80–86.

[8] Natural Language Toolkit - NLTK 3.0 documentation, www.nltk.org, (Accessed: 22 April 2015).

[9] Blei, Ng, and Jordan. 2003, Latent dirichlet allocation. The Journal of Machine Learning Research, pp. 993–1022.

[10] Thomas K. Landauer and Susan Dumais. 2008, Latent semantic analysis. Scholarpedia, 3(11):4356.

[11] Mei, Ling, Wondra, Su and Zhai. 2007 Topic sentiment mixture: modeling facets and opinions in weblogs. In Proceedings of the 16th Int. Conference on World Wide Web, pp. 171–180.

[12] Lin, C. and He, Y. 2009 Joint sentiment/topic model for sentiment analysis. Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375–384.

[13] Li, Huang, and Zhu. 2010 Sentiment Analysis with Global Topics and Local Dependency. AAAI, pp. 1371–1376.

[14] Li, Zhou, Gao, Wong and Wei. 2011 An effective approach for topic-specific opinion summarization. Information Retrieval Technology, pp. 398–409.

[15] Steinberger, J. and Ježek, K. 2004 Using Latent Semantic Analysis in Text Summarization. In Proceedings of ISIM 2004, pp.es 93-100.

[16] Steinberger, J. and Krišťan, M. 2007 Lsa-based multi-document summarization. In Proceedings of 8th International Workshop on Systems and Control, pp. 1–5.

[17] Steinberger, J. and Ježek, K. 2009 Update summarization based on latent semantic analysis. Text, Speech and Dialogue, pp. 77–84.

[18] Steinberger, J. and Ježek, K. 2012 Evaluation measures for text summarization. Computing and Informatics, pp. 1001–1025.