FACULTY
OF APPLIED SCIENCES
UNIVERSITY
OF WEST BOHEMIA

University of West Bohemia
Department of Computer Science and Engineering
Univerzitní 8
306 14 Pilsen

Contrastive Summarization

The State of the Art and Concept of Ph.D. Thesis

Michal Campr

Technical Report No. DCSE/TR-2013-08

September, 2013

Distribution: public

Technical Report No. DCSE/TR-2013-08

September, 2013

Contrastive Summarization

Michal Campr

Abstract

With the continual growth of the internet as a source of information, where a great amount of data is being uploaded every minute, the need for data compression is getting more important. This applies also to textual documents. As the amount of data grows, the probability of multiple similar documents is increasing. One way how to deal with this is to reduce the text, i.e. to use document summarization. In this work, we focus on a specific task – contrastive summarization. The aim is to compare two sets of textual documents in order to obtain latent topics discussed there, as well as the opinions of authors, and create a summary depicting the main topics and opinions mentioned there. The first two chapters of this work describe the basic terms and problems of summarization. The third and fourth chapter sum up the existing methods in comparative and contrastive summarization respectively. The last chapter describes our future plans and our proposal of a novel method for contrastive summarization.

Copies of this report are available on
http://www.kiv.zcu.cz/en/research/publications/
or by surface mail on request sent to the following address:

University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Contrastive Summarization - PhD Thesis Proposal

Author:

Ing. Michal Campr

Supervisor:

Prof. Ing. Karel Ježek CSc.

28. July 2013

Contents

Chapter 1

Introduction

The main objective of our research is the development of a new contrastive summarization algorithm that would take advantage of latest findings in the area of Natural Language Processing (NLP). The resulting method will combine several NLP methods in order to create a contrastive summary of two document groups, which will depict the most important differences between those groups. The resulting summary should take into account the topics of the documents as well as the sentiment (or opinion) of authors.

This section further contains the motivation and the document structure.

1.1 Motivation

With the continual growth of the internet as an important source of information, where a great amount of data is being uploaded every minute, the need for data compression is getting more important. This necessity applies not only for audio or video files, but also for textual documents. As the amount of data grows, the probability of multiple similar documents is increasing (e.g. on sites containing product reviews etc.). One way how to make it easier for people to process so much information is to reduce the text.

However, it is not a simple task to create a shortened version of a document, i.e. a summary, because one has to really understand the ideas expressed in the text. This requires a thorough analysis of the documents structure, latent topics and the authors opinion. In order to create a meaningful summary for the user, it is also appropriate to take into account the other similar documents, so that the reader can compare the viewpoints of others.

1.2 Document structure

This document is organized as follows: The next chapter contains the description of general text summarization and lists several categories in which this area can be divided. The third chapter covers related work in the area of comparative summarization as well as our approach to this particular task using Latent Semantic Analysis (LSA) and Latent Dirichlet allocation (LDA). The task of comparative summarization is discussed here, because it is closely related to contrastive summarization and many of the algorithms used there can be utilized for our problem. The chapter 4 covers related works in the area of contrastive summarization. A variety of algorithms solving this problem is briefly described here. Finally, in the chapter 5, the problem of our future research is outlined and the features of the summarizer we intend to construct are discussed. In the end, the chapter 6 sums up the information provided in this paper and concludes it.

Chapter 2

Document summarization

The term ’document summarization’ is just one of many problems in the NLP area, and it basically deals with document reduction. The main goal of summarization is to reduce the amount of information in a textual document while preserving the most important information. A summarizer can select the information with respect to the user preferences or document structure etc. so there are obviously many different approaches based on their features which are listed below. Note that the following list of summarization tasks is not complete and consists only of those relevant for this paper (the most important are highlighted in bold).

1. Form of the summary

   (a) Abstract - the resulting summary consists especially of newly synthesized sentences. This technique is generally very hard to implement with the use of a computer, because there are other issues which have to be solved besides the plain summarization. For example a sophisticated semantic analysis of the input text and synthesis of the resulting sentences is needed.

   (b) **Extract** - the result is composed of sequences of words from the original text. The most often used method, and proven to work the best in many papers, is selecting sentences, which were assigned the best score.

2. Purpose of the summary
   (a) General - the summary is created without considering any additional parameters.
   (b) Query based - the resulting summary is created with considering an input query from the user.
   (c) Sentiment - the summary considers author's positive or negative sentiment or opinion expressed in a single document.
   (d) Update - the result takes into account some information, which the user already knows, and tries to emphasize any new information from a new set of documents.
   (e) **Contrastive** - the resulting summaries try to highlight the main differences in the sentiment of authors of two different sets of documents.
   (f) Comparative - the result consists of two separate summaries, which sum up the most significant factual differences in two separate sets of documents.
3. Size of input data
   (a) Single-document summarization
   (b) **Multi-document** summarization
4. Language
   (a) Single-language summarization
   (b) **Multi-language summarization**
5. Used method
   (a) Heuristics, statistics - e.g. Naive Bayes
   (b) Graph methods - e.g. PageRank, TextRank, LexRank
   (c) Algebraic methods - e.g. **Latent semantic analysis** (LSA) or NMF (non-negative matrix factorization)

Chapter 3

Comparative summarization

Because of the problem described in 1.1, we already explored the possibilities of utilizing two popular topic models - Latent Semantic analysis (LSA) and Latent Dirichlet Allocation (LDA) for comparative summarization, which is quite a recent area of research and several methods have already been explored. The purpose of these methods is to find information, including some latent information, about the input documents and find factual differences between them. These differences are then represented by the most characteristic sentences which form the resulting summaries.

3.1 Existing approaches to comparative summarization

This section briefly explains methods, that have been already published, addressing the problem of comparative summarization via various techniques.

3.1.1 Comparative document summarization via discriminative sentence selection

Paper [1] proposes a new sentence selection method (based on a multivariate normal generative model) for extracting sentences which represent specific characteristics of multiple document groups. Given a collection of document groups (clusters), the documents are decomposed into a set of sentences F and the sentence-document and sentence-sentence similarities are computed using cosine similarity.

The problem of sentence selection is formalized as selecting a subset of sentences, $S \subset F$, to accurately discriminate the documents in different groups, i.e. to predict the group identity variable Y. Selecting an optimal subset of sentences from documents is considered a combinatorial optimization problem and thus, the best practice is to take a greedy approach, i.e. sequentially selecting sentences to achieve a sub-optimal solution.

3.1.2 Comparative News Summarization Using Linear Programming

In paper [2], a novel approach to generating comparative news summaries is proposed. 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 main task is to extract individual descriptions of each topic over the same aspects and then generate comparisons. To discover latent comparative aspects, a sentence is considered as a bag of concepts. The final summary should contain as many important concepts as possible. An important concept is likely to be mentioned frequently, and thus the frequency is used as a measure of importance. Each concept is represented with the use of words, named entities and bigrams.

The objective function score of a comparative summary can be estimated as:

$$\lambda \sum_{j=1}^{|C_1|} \sum_{k=1}^{|C_2|} u_{jk} \cdot op_{jk} + (1-\lambda) \sum_{i=1}^{2} \sum_{j=1}^{|C_i|} w_{ij} \cdot oc_{ij}, \tag{3.1}$$

where the first component is the estimation of comparativeness and the second is an estimation of representativeness. $\lambda = 0.55$ is a factor that balances comparativeness and representativeness. $C_i = c_{ij}$ is the set of concepts in the document set D_i ($i = 1$ or 2). Each concept c_{ij} has a weight $w_{ij} \in R$. $oc_{ij} \in 0, 1$ is a binary variable indicating whether the concept c_{ij} is present in the summary. A cross-concept pair $< c_{1j}, c_{2k} >$ has a weight $u_{jk} \in R$

and op_{jk} is a binary variable indicating if this pair is present in the summary. The weights are calculated from term frequencies.

The resulting algorithm selects proper sentences to maximize the defined objective function. The optimization of this function is an integer linear programming problem and was solved using the IBM ILOG CPLEX optimizer.

The experiment to verify this method was conducted on five chosen pairs of comparable topics, and for each of them, ten articles were retrieved. The comparative summaries for each topic pair were written manually. The resulting evaluation using ROUGE showed that the proposed model achieved best scores over all metrics.

3.1.3 Multi-document summarization via the minimum dominating set

The paper [3] presents a newly proposed framework for multi-document summarization using the minimum dominating set of a sentence graph which is generated from a set of documents. This framework is constructed to be able to address four well-known summarization tasks including generic, query-focused, update and comparative summarization. There are also proposed approximation algorithms for solving the minimum dominating set problem.

A dominating set of a graph is a subset of vertices such that every vertex in the graph is either in the subset or is adjacent to a vertex in the subset. A minimum dominating set is a dominating set with the minimum size. Many approximation algorithms for finding the minimum dominating set have been developed. It has been shown that this problem is equivalent to the set cover problem, which is a well-known NP-hard problem and an existing greedy algorithm has been chosen for this particular task.

The sentence graph for generating the summary has been generated as follows: each node is a sentence from a document collection; sentences are represented as vectors based on tf-isf (term-frequency, inverted sentence frequency); a cosine similarity is computed for each pair of sentences and if it is above a given threshold, an edge is added between the corresponding nodes. After the graph is constructed, the summarization problem is solved via finding the minimum dominating set.

For comparative summarization, this method is extended to generate the discriminant summary for each group of documents. Given N groups of documents $C_1, C_2, ..., C_N$, the sentence graphs $G_1, G_2, ..., G_N$ are constructed. To generate the summary for $C_i, 1 \leq i \leq N, C_i$ is viewed as the update of all other groups. To extract a new sentence, only the one connected with the largest number of sentences which have no representatives in any groups will be extracted. This extracted set is denoted as the complementary dominating set. To perform comparative summarization, the dominating sets $D_1, D_2, ..., D_N$ are extracted at first. Then the complementary dominating set CD_i is extracted for G_i. And finally, from this set, the summary is constructed.

For evaluating the comparative summarization, a case study for comparing results of various methods was performed.

3.1.4 A cross-collection mixture model for comparative text mining

The paper [4] focuses on a text mining problem, called Comparative Text Mining. The main task is to discover any latent common themes in a set of comparable text collections as well as summarize their similarities and differences. A generative probabilistic mixture model is proposed, which simultaneously performs cross-collection and within-collection clustering.

The Comparative Text Mining in general involves:

- Discovering common themes (topics or subtopics) across all collections of documents.
- For each discovered theme, characterize what is in common among all the collections and what is unique in each of them.

Besides identifying the themes in one collection, there is the need to discover themes across all collections. This task is more challenging, because it involves a discriminative component, and mainly, because there are no training data. This is the reason, why an unsupervised learning method, such as clustering, was used.

For this task, a probabilistic mixture model for clustering, which is closely related to probabilistic latent semantic indexing model, was adapted. In addition to considering k latent common themes across all collections (obtained from the original clustering mixture model), a potentially different set of k collection-specific themes is considered.

The resulting model generates k collection-specific models for each collection and k common theme models across all collections. These models are word distribution or unigram language models. The high probability words can characterize the given theme/cluster and these words can be directly used as a summary or indirectly (e.g. through a hidden Markov model) to extract relevant sentences to form a summary.

This model was evaluated on two different data sets (news articles and laptop reviews) by comparing with a baseline clustering method based on a simple mixture model.

3.1.5 Summarizing similarities and differences among related documents

The main focus of paper [5] is to provide a tool for analyzing document collections such as multiple news stories. This tool can be used to detect and align similar regions of text among individual documents, and to detect relevant differences among them. Given a topic and a pair of related news stories, the resulting method identifies salient regions of each story related to the topic, and then compares them, summarizing similarities and differences. The used method consists of three phases: analysis, refinement and synthesis phase.

The analysis phase consists of extracting words, phrases and proper names and building their graph representation. In particular, nodes represent word instances at different positions, with phrases and names being formed out of words. Associated with each node is a record characterizing the various features of the word in that position, e.g. absolute word position, position in sentence, tf-idf (term-frequency, inversed document frequency) weight etc. Nodes in the graph can have adjacency links to textually adjacent nodes, links to other instances of the same word, links between nodes which belong to a phrase and links that form proper names.

The refinement phase makes use of the relationships between term instances to determine what is salient.

The synthesis phase uses the obtained set of salient items and according to them extracts corresponding text excerpts of the source to form a summary.

For the purpose of finding the differences in a set of documents, graphs $G'_1...G'_n$ (representations of each document), graph C (Commonalities) and D (Differences) need to be constructed. Graph C contains only distinct terms, not term occurrences and is represented as a term-document matrix, where the weight of each distinct term in a document is the highest weight of any of its occurrences in that document, normalized by the maximum weight of any term in that document. Graph D is defined as $D = (G'_1... \cup G'_n) - C$.

There are several strategies on forming the resulting summary:

- Ranking sentences based on weights of contained words and thus skipping computing the Commonalities and Differences. This is a very simple strategy, but does not guarantee that higher-ranked sentences reflect the needed information.

- In cross-document sentence extraction, the best sentences containing words in C/D based on their total weight to separately summarize the commonalities and differences respectively.

- In cross-document sentence alignment, pairs of sentences, one from each document, are ranked for coverage of common words.

- Techniques for extracting fragments instead of sentences. These include ”bag-of-terms” strategies as well as generation of well-formed sentence fragments.

3.2 Summarizing the Differences in Multilingual News

The paper [6], investigates the task of multilingual news summarization for the purpose of finding the main differences between news articles about the same topic in English and

Chinese languages. Two novel graph-based ranking approaches are proposed - CoRank and C-CoRank (Constrained CoRank).

CoRank can extract both Chinese and English summaries from the two document sets in a unified graph-based ranking process. Each sentence was assigned a difference score indicating how much it contains important but differential information. This score relies on both English and Chinese sentences and is based on the following assumptions:

- 1: The difference score of a Chinese sentence would be high if it is heavily correlated with other Chinese sentences with high difference scores in the Chinese documents.
- 2: The difference score of a Chinese sentence would be high if it is very unrelated to the English sentences with high difference scores in the English document set.
- 3: The difference score of an English sentence would be high if it is heavily correlated with other English sentences with high difference scores in the English document set.
- 4: The difference score of an English sentence would be high if it is very unrelated to the Chinese sentences with high difference scores in the Chinese documents.

C-CoRank improves the previous method by adding a new factor (common score) for each sentence. This score indicates how much a sentence contains important and common information and is based on the following additional assumptions:

- 5: The common score of a Chinese sentence would be high if it is heavily correlated with other Chinese sentences with high common scores in the Chinese documents.
- 6: The common score of an English sentence would be high if it is heavily correlated with other English sentences with high common scores in the English documents.
- 7: The common score of a Chinese sentence would be high if it is heavily correlated with the English sentences with high common scores in the English documents.
- 8: The common score of an English sentence would be high if it is heavily correlated with the Chinese sentences with high common scores in the Chinese documents.

- 9: The sum of the difference score and the common score of each sentence is fixed to a particular value.

These two methods were evaluated on manually labeled Chinese and English summaries. The dataset consists of 15 news topics with 36.3 Chinese articles on average per topic and 28.3 English articles on average per topic. For the purpose of the ranking process, both languages were translated to the other with the use of Google Translate online service. In the experiment, the summary length was set to five sentences and the final evaluation was done using the ROUGE-1.5.5 toolkit.

3.3 Our approaches to comparative summarization

In this section, we describe our experiments with using LSA and LDA topic models for comparative summarization. Both methods are firstly explained on simpler task, so to make it more understandable.

3.3.1 Using LSA for Update Summarization

Latent Semantic Analysis (LSA) is an algebraic method, which can analyze relations between terms and sentences of a given set of documents. It uses SVD (Singular Value decomposition) for decomposing matrices. SVD is a numerical process, which is often used for data reduction, but also for classification, searching in documents and for text summarization [7] [8]. Update summarization [9] works with two different sets of documents D_1 and D_2. The assumption is that the user has already read the documents D_1 and wants to get an estimate of what is new in set D_2 from D_1.

The whole process of summarization starts with creating two matrices A_1 and A_2 for each of the document sets. Each column vector of matrix A contains frequencies of terms in sentences. Both matrices must however be created with the same set of terms (terms from both document sets combined) to avoid inconsistencies with singular vector lengths. So the matrix A_1 has $t \times s_1$ dimensions and matrix A_2 has $t \times s_2$ dimensions, where t is the number of terms in both document sets, s_1 is number of sentences in the first set and s_2 is the number of sentences in the second document set. The values of these matrices are computed as $a_{ij} = L(t_{ij}) \cdot G(t_{ij})$, where $L(t_{ij})$ is a boolean value (0 if term i is present in sentence j, 1 otherwise) and $G(t_{ij})$ is the global weight for term i in the whole document:

$$G(t_{ij}) = 1 - \sum_j \frac{p_{ij} log(p_{ij})}{log(n)}, p_{ij} = \frac{t_{ij}}{g_i}, \tag{3.2}$$

where t_{ij} is the frequency of term i in sentence j, g_i is the total number of times that term i occurs in the whole document and n is the number of sentences in the document.

The Singular Value Decomposition of matrix A, constructed over a single document with m terms and n sentences, 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 so called singular values. $V = [v_{ij}]$ is an $n \times n$ matrix and its columns are called right singular vectors. This decomposition provides latent semantic structure of the input document represented by the matrix A. This means, that it provides a decomposition of the document into n linearly independent vectors, which represent the main topics contained in the document. If a specific combination of terms is often present within the document, then this combination is represented by one of the singular vectors. And furthermore, the singular values contained in the matrix Σ represent the significance of these singular vectors (or topics). Matrix U then provides mapping of terms on topics and matrix V provides mapping of sentences on topics.

By applying the SVD decomposition on both matrices A_1 and A_2 separately, we acquire the matrices U_1 and U_2, Σ_1 and Σ_2, V_1^T and V_2^T, which provide the mapping of terms/sentences on topics, contained in both document sets. We can then start comparing those topics contained in matrices U_1 and U_2: for each "new" topic (left singular vector) in U_2, we want to find the most similar topic in U_1. The degree of similarity (redundancy of the topic) between two vectors is computed as a cosine similarity:

$$red(t) = \frac{\sum_{j=1}^{m} U_1[j,i] * U_2[j,t]}{\sqrt{\sum_{j=1}^{n} U_1[j,i]^2}} * \sqrt{\sum_{j=1}^{n} U_2[j,t]^2}, \tag{3.3}$$

where t is the index of the "new" topic from U_2, j is the index of topic from U_1, m is the index of a matrix row. With computed redundancy, we can get the novelty of the given topic: $nov(t) = 1 - red(t)$.

With the values of $nov(t)$ we create a diagonal matrix US (Update Score) and multiply it by the matrix Σ_2 and V^T. The final matrix $F = US * \Sigma_2 * V_2^T$ then contains the novelty, as well as the importance of individual topics, mapped on sentences.

From the final matrix F, we can then start selecting sentences into the final extract. This selection is based on finding the longest sentence vectors. The length s_r of a sentence r is defined as:

$$s_r = \sqrt{\sum_{i=1}^{t} f_r i^2} \tag{3.4}$$

The selected vector is then subtracted from the matrix F, so that the information contained in the sentence is not chosen again. The process of finding the longest vector then continues until the resulting summary reaches a desired length.

Using LSA for Comparative Summarization

The principle of comparative summarization is loosely based on update summarization, but with a few changes. Its goal is the comparison of two different sets of documents D_1 and D_2, where we do not assume any previous familiarity with any of the documents. We just assume, that those two set of documents refer to a similar topic, but contain different information about this topic. The aim is finding the most important differences between these sets.

The process starts by creating two matrices A_1 and A_2. The next step is applying the SVD decomposition on matrices A_1 and A_2 separately and comparing topics in matrices U_1 and U_2 as was described in the previous section, but this time, we make comparisons for both directions. At first, we start finding the most similar topics in U_1 for each topic from U_2, which results in matrix US_2. We then create the final matrix $F_2 = US_2 * \Sigma_2 * V_2^T$. Similarly, matrix F_1 can be created for the opposite direction. The process of finding the best suitable sentence is then similar, i.e. finding the sentence vector with the largest length s_r.

The process of selecting the best suitable sentences is run on both matrices F_1 and F_2, so the final result contains two different extracts, each telling us, what the main differences in the document sets are. During this process, we have to make sure, that we do not select any sentence which is similar to any already selected sentence by using cosine similarity to detect possible similarities between the candidate sentence and already selected sentences.

3.3.2 Comparative summarization via LDA

Latent Dirichlet Allocation has already been utilized in several methods, but to our knowledge it has not yet been used in the context of comparative summarization. The closest problem already addressed is the so called update summarization. It aims to search for information, which newly arise in a series of documents about the same topic. The assumption is that the user is familiar with one document and would like to know what information are additional in another document. We have investigated the already published methods for basic and update summarization using LDA to learn the possibilities of comparing two sets of documents so that we can utilise the best practises to address the problem of comparative summarization.

3.3.3 Basic summarization via LDA

Latent Dirichlet Allocation (LDA) [10] can be basically viewed as a model which breaks down the collection of documents (the importance of document B for the document set is denoted as $P(D_B)$) into topics by representing the document as a mixture of topics with a probability distribution representing the importance of *j-th* topic for document B (denoted as $P(T_j|D_B)$). The topics are represented as a mixture of words with a probability representing the importance of the *i-th* word for the *j-th* topic (denoted as $P(W_i|T_j)$). This model has already been used for basic summarization in several papers. The topic and word probabilities are in each of the below mentioned methods obtained using the Gibbs sampling method. These summarization methods are briefly described in the following paragraphs. In order to shorten the explanations, only some interesting ideas and explanations (for the purpose of this paper)are mentioned.

The paper [11] has presented new algorithms for scoring sentences based on LDA probability distributions. The basic idea is computing the probability of the r-th sentence from probabilities of words and topics (depending on used algorithm):

$$P(S_r|T_j) = \prod_{W_i \in S_r} P(W_i|T_j) * P(T_j|D_B) * P(D_B) \tag{3.5}$$

or

$$P(S_r|T_j) = \frac{\sum_{W_i \in S_r} P(W_i|T_j) * P(T_j|D_B) * P(D_B)}{length(S_r)} \tag{3.6}$$

After obtaining the probabilities $P(S_r|T_j)$, i.e. the probabilities of *r-th* sentence belonging to the *j-th* topic, the selection of the most significant sentences can begin. The process is finished when the number of sentences reaches a predefined amount.

The other paper dealing with LDA-based summarization is [12]. The idea is to combine the LDA topic model and Latent Semantic Analysis (LSA) to reduce the information content in sentences by their representation as orthogonal vectors in a latent semantic space. At first, the LDA probability distributions of topics and words are obtained. After that, for each topic T_j, a term-sentence matrix is created and then the Singular Value Decomposition (SVD) is applied to each of them. The result of the SVD are three new matrices U, Σ and V^T, from which only the third one is utilised. This matrix contains the so called right singular vectors, which basically map topics to sentences. After obtaining the sentence probabilities, the process of selecting sentences with the best score can run until the predefined summary length is reached.

The paper [13] presents two algorithms for summarization and most importantly a new sentence similarity measure based on LDA. Instead of representing a sentence as a sparse vector using tf-idf, the idea is to use the LDA topic model to represent words and sentences as vectors of topic probabilities. The sentence vector is calculated as an average value of topic vectors of all words in the given sentence. Using this representation, it is a simple matter to measure the similarity between any two vectors using cosine similarity. The summarization algorithms are then based on selecting the best candidate sentence which also has the lowest redundancy with the existing summary until the summary length is reached.

3.3.4 Update summarization via LDA

The update summarization is the closest problem to ours, so we explored the used methods of comparing LDA topics. The following paragraphs describe methods of update summarization that have been already published and evaluated.

In the paper [14] a novel update summarization framework was proposed. The topics were extracted from two sets of documents A and B by the means of LDA topic model. The topics were assigned into four different categories:

- emerging – topics that newly arise in B
- activating – topics in both set, but with more emphasis in B
- non-activating – topics in both sets, but not too much discussed in B
- perishing – topics only in A

The correlations between old and new topics were then identified with the use of Pearson product-moment correlation. A novel algorithm (CorrRank) was also developed for ranking sentences with topic correlation so that the best ranked sentences can be iteratively added to the resulting summary.

The method proposed in the paper [15] is derived from TopicSum presented in [16] and the topic model of input documents is restricted to only two topics for each document set. The idea is that one topic in each document contains all the already known facts and the second topic contains all the new information that we want to extract.

3.4 Comparative summarization via LDA

This section will thoroughly describe our novel method for comparative summarization using LDA topic model. Our idea is to use this topic model to represent the documents, compare these topics and select the most significant sentences from the most diverse topics, to form a summary.

The first step is to load the input data from two document sets A and B. The important thing here is that from the perspective of LDA, we treat every sentence as one document. When we have all the sentences from both sets loaded, we can estimate the LDA parameters (the exact reason will be discussed in the last section of this paper) as follows:

- summaryLength = 10sentences
- numberOfTopics = $\sqrt{numberOfSentences}$
- numberOfIterations = 3000
- α = 50/numberOfTopics
- β = 200/numberOfWords

Before we run the Gibbs sampler (we used the implementation JGibbLDA) to obtain the LDA topics, we have to remove the stop-words and perform term lemmatization. This way we are sure that there are no words that carry no useful information. With the parameters set and input text prepared, we can obtain the word-topic distributions for each document set and store them in matrices T_A (topic-word) for the document set A and T_B for B, where row vectors represent topics and column vectors represent words. A very important aspect of writing the distributions into matrices is to ensure that both of them have the same dimensions, i.e. to work as well with the words that appear only in one set and including them also in the second matrix (with zero probability). After this, we can compute topic-sentence matrices U_A and U_B with sentence probabilities (we experimented with two equations):

$$P(S_r|T_j) = \frac{\sum_{W_i \in S_r} P(W_i|T_j)}{length(S_r)^l}, \tag{3.7}$$

or

$$P(S_r|T_j) = \frac{\sum_{W_i \in S_r} P(W_i|T_j) * P(T_j|D_r)}{length(S_r)^l}, \tag{3.8}$$

where $l \in< 0, 1 >$ is an optional parameter to configure the handicap of long sentences. The row vectors represent topics and the columns are sentences. Next step covers the creation of a diagonal matrix *SIM* which contains the information about similarities of topics from both sets. This is accomplished as follows:

- $T_A = [T_{A1}, T_{A2}, ..., T_{An}]^T, T_B = [T_{B1}, T_{B2}, ..., T_{Bn}]^T$, where T_{Ai} and T_{Bi} are row vectors representing topics and n is the number of topics.
- For each T_{Ai} find red_i (redundancy of i-th topic) by computing the largest cosine similarity between T_{Ai} and T_{Bj}, where $j \in< 1..n >$ and storing value $1-red_i$ representing the dissimilarity of i-th topic into matrix *SIM*.

Finally, we create matrices $F_A = SIM * U_A$ and $F_B = SIM * U_B$ combining the probabilities of sentences with the novelty of topics. From these matrices, it is a simple matter to find sentences with the best score and including them in the summary. For better results, it is essential to compare the candidate sentence with already selected sentences to avoid information redundancy (the comparison is also achieved via the cosine similarity). If a sentence is selected, the relevant vector in F_A or F_B is set to *0* in order to remove the information from the matrix. The final result consists of two independent summaries of predefined length, each of which depicts the most significant information, which are specific for one of the compared document set exclusively.

Chapter 4

Contrastive summarization

This chapter sums up and describes the main principles of already published methods which address contrastive summarization. This particular NLP task aims to compare two sets of documents and identify the differences in opinions of their authors.

4.1 Existing approaches to contrastive summarization

4.1.1 Contrastive Summarization: An Experiment with Consumer Reviews

The paper [17] is dealing with a variation of entity centric summarization and aims to summarize information about pairs of different entities. The application of the proposed method is oriented on consumer reviews, where a person considering a purchase wants to see the differences in opinion about the top candidate products without reading all reviews. The goal is to generate contrasting opinion summaries of two products based on their consumer reviews. The model used for summarizing is based on the SAM model. It was primarily used for generating single opinionated summaries by selecting a number of text excerpts so that the summary represents the average opinion and speaks about its important aspects. The SAM model is able to create a probability model over the given set of sentences T with emphasis on their associated sentiment, formalized as $''SAM(T)$.

The SAM summarizer scores each potential summary S by learning the probability model SAM(S). Then it measures the distance between a model learned over the full set of text excerpts T and a summary S by using the KL divergence between their probability distributions. The final summary is the one with the maximal KL divergence. This method can be used for generating contrastive summaries. As input are assumed two products x and y as well as corresponding sets of opinions Tx and Ty. As output, two summaries Sx and Sy, are produced highlighting the differences in opinion between two products. To achieve the best result, the KL divergence is measured between the summaries and the source sets of opinions. The resulting score between two summaries is defined as:

$$\begin{aligned} L(S_x, S_y) = &- KL(SAM(T_x), SAM(S_x)) - KL(SAM(T_y), SAM(S_y)) \\ &+ KL(SAM(T_x), SAM(S_y)) + KL(SAM(T_y), SAM(S_x)) \end{aligned} \tag{4.1}$$

4.1.2 Generating Comparative Summaries of Contradictory Opinions in Text

In the paper [18], authors are dealing with a novel summarization problem called contrastive opinion summarization (COS). Given two sets of positively and negatively opinionated sentences which are the outputs of an existing opinion summarizer, COS aims to extract comparable pairs of sentences representing both positive and negative opinions. The problem is formulated as an optimization problem and two different approximation methods are proposed.

Two sentence similarity functions are needed:

- $\Phi(s_1, s_2) \in < 0, 1 >$ - this content similarity function is to be used to measure the similarity of two sentences in the same opinion group.
- $\Psi(u, v) \in < 0, 1 >$ - this contrastive similarity function measures how well two sentences from opposite opinion groups match up with each other.

Two similarity functions for comparing candidate summaries are proposed:

- $r(S)$ (Representativeness) - measures how well the summary S represents the opinions expressed in the source sets of sentences.
- $c(S)$ (Contrastiveness) - defined as average contrastive similarity Y of the sentence pairs in S.

The first approximation method is Representativeness-first Approximation and is designed to optimize representativeness in the first place by selecting k sentences from each opinion set that best represent all sentences. This method uses a clustering algorithm (hierarchical agglomerative clustering algorithm) to generate k clusters for each opinion set, then taking the most representative sentence from each cluster. The next step is optimizing contrastiveness by aligning the clusters from both opinion sets into the right order, so that every pair of clusters with the same index has the highest contrastiveness.

The second approach is Contrastiveness-first Approximation. At first, contrastive similarity is computed for all pairs of sentences, where the sentences are from opposite opinion groups. These pairs are then sorted in descending order. The following steps of selecting pairs of sentences into the final summary also include computing their representativeness and according to its value deciding which pair to select next.

4.1.3 Sentiment Summarization: Evaluating and Learning User Preferences

Paper [19] presents the results of a large-scale, end-to-end human evaluation of various sentiment summarization models: Sentiment Match (SM), Sentiment Match + Aspect Coverage(SMAC), Sentiment Aspect Match (SAM).

The first system (SM) attempts to extract sentences so that the average sentiment of the summary is as close as possible to the entity level sentiment. Thus, the model prefers summaries with average sentiment as close as possible to the average sentiment across all the reviews. There is an obvious problem with this model. For entities that have a mediocre rating, i.e., $R \approx 0$, the model could prefer a summary that only contains sentences with no opinion whatsoever. This was addressed by prohibiting the algorithm from including a given positive or negative sentence in the summary if another more positive/negative sentence is not included. Thus the summary is forced to consist of only the most positive and most negative sentences, the exact mix being dependent upon the overall star rating.

The SMAC system attempts to model diversity by building a summary that trades-off maximally covering important aspects with matching the overall sentiment of the entity. This system has its roots in event-based summarization for the news domain, where an optimization problem was developed that attempted to maximize summary informativeness while covering as many (weighted) sub-events as possible.

The last system SAM attempts to cover important aspects, but also cover them with appropriate sentiment. A probabilistic approach was employed as it provided performance benefits based on development data experiments. Under the SAM model, each sentence is treated as a bag of aspects and their corresponding mentions' sentiments.

The evaluation of reviews for 165 electronic products (each at least with 4 and up to 3000 reviews) shows that users have a strong preference for summarizers that model sentiment over non-sentiment baselines, but have no broad overall preference between any of the sentiment-based models. However, an analysis of the human judgments suggests that there are identifiable situations where one summarizer is generally preferred over the others. This fact was exploited to build a new summarizer by training a ranking SVM model over the set of human preference judgments that were collected during the evaluation, which resulted in a 30% relative reduction in error over the previous best summarizer.

4.1.4 An exploration of sentiment summarization

Paper [20] introduced the idea of a sentiment summary, a single passage from a document that captures an authors opinion about his or her subject. Using supervised data from the Rotten Tomatoes website (3897 full-text movie reviews), authors examined features that appeared to be helpful in locating a good summary sentence, such as:

- Location of quotations within paragraph - Quotations occur most often at the ends of paragraphs - 47.6% begin at the start of a paragraph, while 26.1% conclude at the end.
- Location of quotations within Document
- Word Choice - The words that appear more frequently in quotations often express emotion directly. Words that are interchangeable with ”movie” are also more common, as are several other words with varied meanings. In addition to words, formatting is a useful predictor. Italicized words and phrases (such as titles) make 8.9% (893 of 10152) of their appearances in quotations, while parentheses make only 2.9%

The sentiment summarization as approached as a classification problem at the sentence level. The mentioned features are used to fit Naive Bayes and regularized logistic regression models for summary extraction.

4.1.5 Summarizing Opinions in Blog Threads

An approach to summarizing positive and negative opinions in 51 downloaded blog threads was presented in paper [21]. First, a sentiment analysis system was applied, which divided the input sentences into three groups: sentences containing positive sentiment, sentences containing negative sentiment and neutral or objective sentences. In order to have a more extensive database of affect-related terms, WordNet Affect, SentiWordNet and MicroWNOp databases were used. Each of the employed resources were mapped to four categories, which were given different scores: positive (1), negative (-1), high positive (4) and high negative (-4).

First, the score of each of the blog post was computed as sum of the values of the words identified; a positive score leads to the classification of the post as positive, whereas a final negative score leads to the system classifying the post as negative.

Then the positive and the negative sentences were passed on to a standard LSA-based summarization system separately to produce one summary for the positive posts and another one for the negative ones.

4.1.6 Thumbs up? Sentiment Classification using Machine Learning Techniques

In paper [22] was considered the problem of classifying documents not by topic, but by overall sentiment, e.g., determining whether a review is positive or negative. Using movie reviews (downloaded from the Internet Movie Database - IMDb) as data was found that standard machine learning techniques definitively outperform human-produced baselines. However, the three machine learning methods that were employed do not perform as well on sentiment classification as on traditional topic-based categorization.

The aim of this experiment was to examine whether it suffices to treat sentiment classification simply as a special case of topic-based categorization (with the two "topics" being positive sentiment and negative sentiment), or whether special sentiment-categorization methods need to be developed. The authors experimented with three standard algorithms: Naive Bayes classification, maximum entropy classification, and support vector machines. The philosophies behind these three algorithms are quite different, but each has been previously shown to be effective in text categorization studies.

The results produced via machine learning techniques are quite good in comparison to the human-generated baselines discussed. In terms of relative performance, Naive Bayes tends to do the worst and SVMs tend to do the best, although the differences aren't very large. On the other hand, the authors were not able to achieve accuracies on the sentiment classification problem comparable to those reported for standard topic-based categorization, despite the several different types of features they tried.

4.1.7 A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts

Authors of paper [23] proposed a novel 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; this greatly facilitates incorporation of cross-sentence contextual constraints.

The document-level polarity classification can be considered to be just a special case of text categorization with sentiment rather than topic-based categories. Hence, standard machine-learning classification techniques, such as support vector machines (SVMs), can be applied to the entire documents themselves, as was done in [4]. The authors refer to such classification techniques as default polarity classifiers. However, the polarity classification can be improved by removing objective sentences (such as plot summaries in a movie review). Therefore, it was proposed to first employ a subjectivity detector that determines whether each sentence is subjective or not: discarding the objective ones creates an extract that should better represent a reviews subjective content to a default polarity classifier.

Chapter 5

Our future research

This chapter contains a description of the final goal of our research and thus my PhD thesis (section 5.1), as well as a plan of our works in section 5.5. Also, a rough proposal of the expected algorithm is covered in section 5.3.

5.1 The aim of the future work

The ultimate goal of our future work is to construct a summarizer which will analyze the input documents and create a contrastive summary. This summary should contain the most important information about the discussed topics and especially taking into account the opinions of the authors. These summaries can be very useful, e.g. for people deciding which computer (or any other product) to buy, because they should contain both positive and negative opinions about the most often discussed topics, such as screen quality or battery life.

The term ′contrastive summarizer′ is a very general label for our planned software and it contains several options to be explored and examined. For example, we must first determine the form of output we would like to create, and so the particular way of dealing with multiple documents. There are basically two possible ways, which we intend to explore:

- Comparing two sets of documents similarly as we did in our work with comparative summarization and upgrading our current system with algorithms for sentiment analysis. The output of such a system would be two summaries, each depicting topics (or aspects) from both sets that are discussed the most in each set, and are also the topics on which the authors have different opinions.

- Treating the whole dataset as one complete document and search for the most common topics discussed here. In the result, each topic would be represented by a pair of sentences which depict the differences in opinions about that particular topic, hence creating a structured contrastive summary.

5.2 Summarizer features

The resulting summarizer will utilize several NLP methods to provide a thorough document analysis, allowing us to compare the input texts and score their sentences accordingly to the topic they discuss and also to the authors opinions.

The features of the summarizer will possibly include the following: sentence boundary disambiguation, term recognition, lemmatization, synonymy recognition, coreference resolution, multiword expression recognition, latent topic recognition, sentiment analysis and sentence scoring.

Many of these tasks are planned to be solved with the help of the Natural Language Toolkit (NLTK 2.0) written in Python. This package offers several algorithms for some of the most common NLP tasks, such as sentence disambiguation or term recognition, so we intend to test these algorithms and utilize them in our summarizer. However, a problem arises if we intend to use our summarizer for other languages. Because of this, we intend to build our algorithms to work with english first and after we test and evaluate it, we will start to adapt it to work with other languages,.

5.3 Summarizer algorithm proposal

The ultimate goal of the summarizer's extracting algorithm is to select sentences with highest informative value, so the task of sentence disambiguation is crucial. So the first step of the algorithm is to divide the input string, which can consist of single or multiple documents, into sentences which can be scored depending on their further analysis. Though being an important step, this problem has been explored many times and so we intend to utilize the mentioned NLTK library.

The next step is to divide each sentence into words and using lemmatization to determine individual terms appearing in that sentence. We have already successfully used a lexicon based lemmatization in our work on comparative summarization, so no further research in this area is planned.

We also plan to experiment with adding other features to the previous step, such as a simple lexicon based algorithm for synonymy and also multiword expression recognition. Furthermore, another interesting addition to this step can be incorporating a deeper analysis of sentences aimed at coreference resolution, which would result into better term recognition, e.g. the word 'he' could be replaced by a particular name and thus not be removed as a stop-word as it usually is. However, because these features are still being extensively worked on by other researchers, we do not intend to conduct any research in these areas and only add them to our summarizer, if suitable algorithms are available. For example, the task of multiword expression recognition is being researched by our colleagues in our Textmining Research Group (http://textmining.zcu.cz/).

A crucial feature will be the analysis of sentiment [24], together with obtaining the latent topics. The problem of joint sentiment-topic modeling, based on LDA, has already been explored ([25], [26], [27], [28]), however they are focusing only on document-level opinions and thus, in order to solve our problem (contrastive summarization), which operates on the sentence-level and needs to compare them based on topics and opinions, further research in the area of joint topic and opinion modeling is needed. For comparing topics, we already experimented with LDA and LSA and from those two topic models, LSA came as a better choice, however, to our knowledge, there has not been any research on how to incorporate sentiment analysis with LSA topic model, so this will be the first and the main problem, which we intend to look into.

The last step is using our joint sentiment-topic model to score sentences so that the best candidates can be included into the final summary. Also, the cosine similarity will be used to compare candidate sentences with already selected ones in order to make sure that any similar sentences will not be selected. This will offer a chance for sentences with other information to be included.

5.4 Dataset for testing and evaluation

There are many summarization methods and also algorithms dealing with sentiment analysis, but to our knowledge there is no unified testing dataset for these two areas combined. Many papers dealing with these problems conducted their evaluation on data downloaded from various websites depending on the particular task. Some were focused on analyzing Twitter messages, others on political issues (bitterlemons.net) or product reviews (e.g. from amazon.com).

Our intention is to focus on product reviews or news articles, because they tend to be much longer than for example Twitter messages, which are limited only to 140 characters per message. And also because the language used there is more formal and thus easier to analyze than messages from any social media.

Our current plan involves manual examination of several product reviews and news article sources, such as Europe Media Monitor (http://emm.newsbrief.eu/), WikiNews (http://en.wikinews.org/) or Project Syndicate (http://www.project-syndicate.org/). After choosing the best suitable source and downloading articles, we intend to manually create our own dataset for evaluating purposes based on these articles.

5.5 Work plan

My PhD studies are planned to last 4 semesters and so we chose to solve the main subtasks in four steps and also published in four papers:

1. The first step involves creating our joint sentiment-topic model based on Latent Semantic Analysis, as was described in section 5.3. We plan to explore the possibilities of joining sentiment analysis with LSA topic model, which we used for comparative summarization before, and publish our results during the first semester.

2. A crucial step of our work is creating a new corpus (section 5.4) for testing and evaluating our novel sentiment-topic model mentioned in the previous step. The process of creation is expected to last longer, because it will involve manual work of several annotators, and thus is planned to be started right in the beginning of our works. The final result is planned to be published during the second semester.

3. As was mentioned in section 5.3, several joint sentiment-topic models have already been explored, so the next step of our research is to evaluate our novel sentiment-topic model on our newly created corpus and compare the results with other works in the area. This evaluation should be completed and published during the third semester.

4. The final and most important step is using our sentiment-topic model for contrastive summarization, as was already described in section 5.3. Completing and evaluating this step is planned to be finished and published at the end of the last semester, along with finishing my PhD thesis.

Chapter 6

Conclusion

This work is discussing two particular NLP problems from the area of text summarization - comparative and contrastive summarization. Although our future research is focused on contrastive summarization, it is appropriate to describe the problem of comparative summarization as well, because it has many similar features and some of the algorithms can be utilized.

We then presented methods, that have been already published, dealing with both mentioned tasks, as well as our approaches to comparative summarization using LSA and LDA. Finally, we outlined the problem which we will be dealing with in the near future and discussed all the features needed for our summarizer and also some optional features that would greatly increase its performance.

Bibliography

[1] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Comparative document summarization via discriminative sentence selection. *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, page 1963, 2009.

[2] X Huang, Xiaojun Wan, and J Xiao. 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, 2011.

[3] Chao Shen and Tao Li. Multi-document summarization via the minimum dominating set. *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984—-992, 2010.

[4] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. A cross-collection mixture model for comparative text mining. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 743—-748, 2004.

[5] Inderjeet Mani and Eric Bloedorn. Summarizing similarities and differences among related documents. *Information Retrieval*, 67:35–67, 1999.

[6] Xiaojun Wan, Houping Jia, Shanshan Huang, and Jianguo Xiao. Summarizing the differences in multilingual news. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 735—-744, 2011.

[7] J Steinberger and M Krišt'an. Lsa-based multi-document summarization. *In Proceedings of 8th International Workshop on Systems and Control*, pages 1–5, 2007.

[8] J Steinberger and Karel Ježek. Using Latent Semantic Analysis in Text Summarization. *In Proceedings of ISIM 2004*, pages 93—-100, 2004.

[9] Josef Steinberger and K Ježek. Update summarization based on latent semantic analysis. *Text, Speech and Dialogue*, pages 77–84, 2009.

[10] DM Blei, AY Ng, and MI Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[11] Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation based multi-document summarization. *Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 91—-97, 2008.

[12] Rachit Arora and B Ravindran. Latent dirichlet allocation and singular value decomposition based multi-document summarization. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 713—-718, 2008.

[13] Tiedan Zhu and Kan Li. The Similarity Measure Based on LDA for Automatic Summarization. *Procedia Engineering*, 29:2944–2949, January 2012.

[14] Lei Huang and Yanxiang He. CorrRank : Update Summarization Based on Topic. *Proceedings of the Advanced intelligent computing theories and applications, and 6th international conference on Intelligent computing*, pages 641–648, 2010.

[15] JY Delort and Enrique Alfonseca. DualSum: a Topic-Model based approach for update summarization. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 214–223, 2012.

[16] A Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. *The 2009 Annual Conference of the North American Chapter of the ACL*, pages 362–370, 2009.

[17] Kevin Lerman and R McDonald. Contrastive summarization: an experiment with consumer reviews. *Proceedings of Human Language Technologies: The 2009 Annual Con-*

ference of the North American Chapter of the Association for Computational Linguistics, pages 113–116, 2009.

[18] HD Kim and CX Zhai. Generating comparative summaries of contradictory opinions in text. *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009.

[19] Kevin Lerman. Sentiment summarization: Evaluating and learning user preferences. *In Proceedings of EACL*, 2009.

[20] Philip Beineke, Trevor Hastie, Christopher Manning, Shivakumar Vaithyanathan, and San Jose. An exploration of sentiment summarization. *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2003.

[21] Alexandra Balahur and MA Kabadjov. Summarizing Opinions in Blog Threads. *PACLIC*, pages 606–613, 2009.

[22] Bo Pang, Lillian Lee, and S Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 10(July):79–86, 2002.

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

[24] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, May 2012.

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

[26] Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Sentiment Analysis with Global Topics and Local Dependency. *AAAI*, pages 1371–1376, 2010.

[27] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and CX Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. *In Proceedings of the 16th Int. Conference on World Wide Web*, pages 171—-180, 2007.

[28] Binyang Li, Lanjun Zhou, Wei Gao, KF Wong, and Zhongyu Wei. An effective approach for topic-specific opinion summarization. *Information Retrieval Technology*, pages 398–409, 2011.

Appendix

Author's previous work

- Michal Campr and Karel Ježek. Comparative summarization via Latent Semantic Analysis. *Proceedings of the 1st WSEAS International Conference on Information Technology and Computer Networks (ITCN '12)*, pages 279-284, 2012
- Michal Campr and Karel Ježek. Comparative Summarization via Latent Dirichlet Allocation. *Proceedings of the DATESO 2013 Workshop*, pages 80-86, 2013
- Michal Campr and Karel Ježek. *Topic Models for Comparative Summarization*. *Proceedings of The 16th International Conference TSD 2013*, pages 568-574, 2013