Comparative summarization via Latent Semantic Analysis

MICHAL CAMPR, KAREL JEŽEK
Department of Computer Science and Engineering
University of West Bohemia
Univerzitní 8, 306 14, Plzeň
CZECH REPUBLIC
mcampr@kiv.zcu.cz, jezek_ka@kiv.zcu.cz   http://textmining.zcu.cz

Abstract: - The primary focus of this paper is multi-document comparative summarization. At first, the concept of comparative summarization is defined, and then the existing approaches are described. Finally, a new method using LSA (Latent Semantic Analysis) for comparative summarization is proposed.

Key-Words: - Comparative summarization, contrastive summarization, summarization via LSA

1 Introduction

The term “automatic summarization” is a general label for one of many problems in the text processing area. There are several different approaches to automatic summarization based on their features, such as the purpose of the summary, the form (abstract or extract), the size and language of the input data or what method is used to generate the summary.

This paper does not focus on the whole area of automatic text summarization, but on one specific summarization task - comparative summarization. Existing methods are briefly presented here and also our novel approach, using latent semantic analysis, to comparative summarization is proposed.

Although this paper focuses only on comparative summarization, we feel that a clear distinction between two very similar tasks - contrastive and comparative summarization – is needed to avoid possible confusion.

Even though these two summarization tasks seem very similar, they have one key aspect in which they differ. Comparative summarization should focus on extracting only differences in topics between pairs of documents and should not consider any sentiment of the author. Finding the differences in sentiment is the main focus of contrastive summarization and it can be used for example to seek out the main opinions about the candidate products the user wants to purchase.

For example, the paper [1] is dealing with a variation of entity-centric summarization and aims to summarize information about pairs of different entities. The application is oriented on consumer reviews, where a person considering a purchase wants to see the differences in opinion about the top candidate products. The goal is to generate contrasting opinion summaries of two products based on their consumer reviews.

In short, the key distinction is in finding differences in topics (comparative summarization) or differences in sentiment (contrastive summarization).

2 Existing Methods of Comparative Summarization

Paper [2] 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 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.

In paper [3], 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:

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

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 $(i = 1 \text{ or } 2)$. 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.

The paper [4] 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 presented in [5] 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.

The paper [6] 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. The results showed that the proposed method is quite effective and performs significantly better than the baseline model.

The main focus of paper [7] 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, thus highlighting what information should be included into the summary.

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 $G_1'\ldots G_n'$ (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 $D = (G_1'\ldots \cup G_n' - C.words$.

With the graphs computed, there are several strategies on forming the resulting summary:

- Ranking sentences in each document 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 Comparative Summarization via Latent Semantic Analysis

This chapter will thoroughly describe LSA as a tool for comparative summarization which is a novel method and our current primary focus. We used this method because we already have experience with its application for update summarization presented in [8].

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.

As was described in [8], update summarization 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 $t\times s_2$ 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 $a_{ij} = L(t_{ij}) \,.\, G(t_{ij})$, 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:

$$G(t_{ij}) = 1 - \sum_j \frac{p_{ij}\log(p_{ij})}{\log(n)}, p_{ij} = \frac{t_{ij}}{g_i}, \quad (2)$$

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 σ. $V = [v_{ij}]$ 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:

$$\text{red}(t) = \frac{\sum_{j=1}^{m} U_1[j,i] * U_2[j,t]}{\sqrt{\sum_{j=1}^{m} U_1[j,i]^2} * \sqrt{\sum_{j=1}^{m} U_2[j,t]^2}}, \quad (3)$$

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:

$$s_r = \sqrt{\sum_{i=1}^{t} f_{ri}^2}. \quad (4)$$

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.

3.2 Using LSA for Comparative Summarization

The principle of comparative summarization is loosely based on the update summarization, described in the previous chapter, 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 in the same way, i.e. by creating two matrices A_1 and A_2. The next step is almost identical: we apply the SVD decomposition on matrices A_1 and A_2 separately and start comparing topics in matrices U_1 and U_2 as was described in chapter 4.1, 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 $F_2 = US_2 * \Sigma_2 * V_2^T$. Similar 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. We have tested three different solutions:

- Setting values of selected vector to 0. This is the simplest solution and it guarantees that once a sentence was selected, it will not be selected again.
- Solution described in [8] – subtracting the selected vector from the final matrix *F*. This removes the selected sentence (and information it contains) from the whole matrix.
- Using cosine similarity to detect possible similarity between the candidate sentence and already selected sentences. This serves the same purpose as the second solution, but does not make any alterations to the final matrix *F*.

3.2 The Experiment

The main problem for evaluating the quality of comparative summarization is that there are currently no testing data available. This is the reason, why we have conducted a simple experiment with data from TAC 2011 conference to find out if the proposed method works as intended.

The available data consist of 100 news articles in total, divided into 10 topics, 10 articles each. With these articles, we have created 720 different pairs of sets of documents (articles) by combining different topics. In every pair, there is one identical topic present in both sets and one topic for each of the sets that are different. This has a simple purpose: to simulate two sets of documents which have something in common, but also some differences.

The experiment consists of eight different configurations of the summarizer, i.e. combining three specific parts of the algorithm:

- Σ
  - 0 - Not including the singular values in the process of creating the matrix F.
  - 1 - Including the singular values in the process of creating the matrix F.
- Comparison
  - 0 - Setting the values of the selected sentence vector to 0.
  - 1 - Using cosine similarity to detect similarities in selected sentences.
- Selection
  - Len - Using vector length for selecting sentences.
  - MI - Searching for a sentence vector which contains maximal value in the matrix F.

Each of the configurations was used to compute summaries of the 720 mentioned combinations of articles. Summary length was set to 10 sentences. The following table contains average number of sentences that were correctly selected (sentences representing the differences). E.g. in the first configuration and computed on all of the 720 combinations – 8.116 sentences on average out of 10 were correctly selected.

An interesting fact can be observed from the final results in Table 1: the precision is generally lower when the “importance” of topics (Σ) is considered. This can be simply explained by the fact, that taking Σ into account changes the matrix *F* in such a way, that the “contrastiveness” of topics is lowered and thus a smaller number of correct sentences is selected. Also, the method of selecting sentences based on maximal value in matrix F comes out as generally the best solution.

Table 1: Results of the experiment

Σ	Comparison	Selection	Precision
0	0	Len	81,16%
		MI	98,44%
	1	Len	81,16%
		MI	98,44%
1	0	Len	61,23%
		MI	84,82%
	1	Len	61,23%
		MI	84,82%

4 Conclusion

Several different approaches have been taken to address the comparative and contrastive summarization problem, but every solution was tested on different data and with a different testing method. This means, that we are not able to compare the results of another comparative summarization technique. This issue is worthy of further attention and could result in some interesting conclusions regarding the comparison of usefulness and performance of the proposed approaches.

Our approach is taking advantage of an already verified method and builds upon it to address the problem of comparative summarization. However, the verification of the results is not yet complete. Although in the Table 1 are presented some interesting conclusions, regarding the precision of selecting sentences depending on used parameters, it does not evaluate the quality of the resulting summary, i.e. if the selected sentences correspond with sentences a human would select. This evaluation is currently our focus and will be completed in the near future. We intend to utilize the ROUGE evaluating method to compare the automatically generated summaries to summaries created by human. When this is done, we should be able to evaluate the quality of our method.

References:

[1] Lerman, K., McDonald, Ryan T. Contrastive Summarization: An Experiment with Consumer Reviews. 2009, HLT-NAACL (Short Papers), pp.113~116

[2] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Comparative document summarization via discriminative sentence selection. In *Proceedings of the 18th ACM conference on Information and knowledge management* (CIKM '09). ACM, New York, NY, USA, 1963-1966.

[3] Xiaojiang Huang, Xiaojun Wan, and Jianguo Xiao. 2011. Comparative news summarization using linear programming. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2* (HLT '11), Vol. 2. Association for Computational Linguistics, Stroudsburg, PA, USA, 648-653

[4] Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics* (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 984-992

[5] Johnson, D.S. 1973. Approximation algorithms for combinatorial problems. *In Proceedings of STOC.*

[6] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. 2004. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '04). ACM, New York, NY, USA, 743-748.

[7] Mani, I., Boedorn, E. 1999. Summarizing Similarities and Differences Among Related Documents. *Inf. Retr.* 1, 1-2 (May 1999), 35-67

[8] Steinberger, J., Ježek, K. Update Summarization Based on Latent Semantic Analysis. In *Text, Speech and Dialogue.* Berlin: Springer, 2009. s. 77-84.