Topic-based Multi-Document Summarization with Probabilistic Latent Semantic Analysis

Leonhard Hennig
DAI Labor, TU Berlin
Berlin, Germany
leonhard.hennig@dai-labor.de

Abstract

We consider the problem of query-focused multi-document summarization, where a summary containing the information most relevant to a user's information need is produced from a set of topic-related documents. We propose a new method based on probabilistic latent semantic analysis, which allows us to represent sentences and queries as probability distributions over latent topics. Our approach combines query-focused and thematic features computed in the latent topic space to estimate the summary-relevance of sentences. In addition, we evaluate several different similarity measures for computing sentence-level feature scores. Experimental results show that our approach outperforms the best reported results on DUC 2006 data, and also compares well on DUC 2007 data.

Keywords

text summarization, probabilistic latent semantic analysis, plsa

1 Introduction

Automatically producing summaries from large textual sources is an extensively studied problem in IR and NLP [17, 12]. In this paper, we investigate the problem of multi-document summarization, where a summary is created from a set of related documents and optionally fulfills a specific information need of a user. In particular, we focus on generating an extractive summary by selecting sentences from a document cluster [8]. Multi-document summarization is an increasingly important task: With the rapid growth of online information, and many documents covering the same topic, the condensation of information from different sources into an informative summary helps to reduce information overload. Automatically created summaries can either consist of the most important information overall (generic summarization) or of the information most relevant with respect to a user's information need (query-focused summarization).

A major aspect of identifying relevant information is to find out what a text is about. A document will generally contain a variety of information centered around a main theme, and covering different aspects of the main topic. Similarly, human summaries tend to cover different topics of the original source text to increase the informative content of the summary. Various approaches have exploited features based on the identification of topics (or thematic foci) to construct generic or query-focused summaries. Often, thematic features rely on identifying and weighting important keywords [21], or creating topic signatures [14, 10]. Sentences are scored by combinations of keyword scores, or by computing similarities between sentences and queries. Yet it is well known that term matching has severe drawbacks due to the ambivalence of words and to differences in word usage and personal style across authors. This is especially important for automatic summarization, as summaries produced by humans may differ significantly, potentially not sharing very many terms [16].

Latent Semantic Indexing (LSI) is an approach to overcome these problems by mapping documents to a latent semantic space, and has been shown to work well for text summarization [9, 23]. However, LSI has a number of drawbacks, namely its unsatisfactory statistical foundations. The technique of probabilistic latent semantic analysis (PLSA) assumes a latent lower dimensional topic model as the origin of observed term co-occurrence distributions, and can be seen as a probabilistic analogue to LSI [11]. It has a solid statistical foundation, is based on the likelihood principle and defines a proper generative model for data. PLSA models documents as a list of mixing proportions for mixture components that can be viewed as representations of "topics" [4].

In this paper, we are primarily interested the capability of the PLSA approach to model documents as mixtures of topics. Unlike previous approaches in PLSA-based extractive summarization, we represent sentences, queries, and documents as probability distributions over topics. We train the probabilistic model on the term-sentence matrix of all sentences in a document cluster, and proceed by folding queries, document titles and cluster centroid vectors into the trained model. This allows us to compute various thematic and query-focused similarity measures, as well as redundancy measures, in the space of latent topics, in order to estimate the summary-worthiness of sentences.

Our system improves on previous approaches in three ways: First, we investigate PLSA in the context of multi-document summarization, modeling topic distributions across documents and taking into account information redundancy. Second, we do not only pick sentences from topics with the highest likelihood in the training data as in [3], but compute a sentence's score based on a linear function of query-focused and

thematic features. Third, we examine how a PLSA model can be used to represent documents, sentences and queries in the context of multi-document summarization, and investigate which measures are most useful for computing similarities in the latent topic space. We evaluate our approach on the data sets of the DUC 2006 and DUC 2007 text summarization challenges, and show that the resulting summaries compare favorably on ROUGE metrics with those produced by existing state-of-the-art summarization systems.

The rest of this paper is organized as follows: In Section 2 we describe the probabilistic latent semantic analysis algorithm. Next, in Section 3, we give details of our summarization system, the sentence-level features we use, as well as of the similarity measures we evaluate. In Section 4, we give experimental results showing that our approach leads to improvements over a LSI baseline, and that overall scores compare well with those of existing systems on ROUGE metrics. We then compare our system to related work in Section 5, and finally Section 6 concludes the paper.

2 Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis is a latent variable model for co-occurrence data which has been found to provide better results than LSI for term matching in retrieval applications [11]. It associates an unobserved class variable $z \in \mathcal{Z} = \{z_1, \ldots, z_k\}$ with each observation (d, w), where word $w \in \mathcal{W} = \{w_1, \ldots, w_i\}$ occurs in document $d \in \mathcal{D} = \{d_1, \ldots, d_j\}$. Each word in a document is considered as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of latent topics. A document is represented as a list of mixing proportions for the mixing components, i.e. it is reduced to a probability distribution over a fixed set of latent classes.

In terms of a generative model, PLSA can be defined as follows:

- select a document d with probability $P(d)$,
- pick a latent class z with probability $P(z|d)$,
- generate a word w with probability $P(w|z)$.

For each observation pair (d, w) the resulting likelihood expression is:

$$P(d,w) = P(d)P(w|d), \text{ where} \quad (1)$$

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d). \quad (2)$$

A document d and a word w are assumed to be conditionally independent given the unobserved topic z. Following the maximum likelihood principle, the mixing components and the mixing proportions are determined by the maximization of the likelihood function

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d,w) log P(d,w), \quad (3)$$

where $n(d, w)$ denotes the term frequency, i.e. the number of times w occurred in d.

The standard procedure for maximizing the likelihood function in the presence of latent variables is the Expectation Maximization (EM) algorithm. EM is an iterative algorithm where each iteration consists of two steps, an expectation step where the posterior probabilities for the latent classes z are computed, and a maximization step where the conditional probabilities of the parameters given the posterior probabilities of the latent classes are updated. Alternating the expectation and maximization steps, one arrives at a converging point which describes a local maximum of the log likelihood. The output of the algorithm are the mixture components, as well as the mixing proportions over the components for each training document, i.e. the conditional probabilities $P(w|z)$ and $P(z|d)$. For details of the EM algorithm and its application to PLSA, see [11].

3 Topic-based summarization

Our approach for producing a summary consists of three steps: First, we associate sentences and queries with a representation in the latent topic space of a PLSA model by estimating their mixing proportions $P(z|d)$[1]. We then compute several sentence-level features based on the similarity of sentence and query distributions over latent topics. Finally, we combine individual feature scores linearly into an overall sentence score to create a ranking, which we use to select sentences for the summary. We follow a greedy approach for selecting sentences, and penalize candidate sentences based on their similarity to the partial summary.

3.1 Sentence representation in the latent topic space

Given a corpus $\mathcal{D}$ of topic-related documents, we perform sentence splitting on each document using the NLTK toolkit[2]. Each sentence is represented as a bag-of-words $\mathbf{w} = (w_1, \ldots, w_m)$. During preprocessing, we remove stop words, and apply stemming using Porter's stemmer [22]. We discard all sentences which contain less than $l_{min} = 5$ or more than $l_{max} = 20$ content words, as these sentences are unlikely to be useful for a summary [24]. We create a term-sentence matrix TS containing all sentences of the corpus, where each entry $TS(i, j)$ is given by the frequency of term i in sentence j. We then train the PLSA model on the term-sentence matrix TS.

After the model has been trained, it provides a representation of the sentences as probability distributions $P(z|s)$ over the latent topics z. This representation can be interpreted as follows: Since the source documents cover multiple topics related to a central theme, each sentence can be viewed as representing one or more of these topics. By applying PLSA, we arrive at a representation of sentences as a vector in

[1] From hereon, we will use $P(z|s)$ and $P(z|q)$ to denote topic distributions over sentences and queries respectively, but for all purposes these can be considered identical to the notation $P(z|d)$ of the original PLSA model.

[2] http://nltk.org

the "topic-space" of the document cluster $\mathcal{D}$:

$$P(z|s) = (p(z_1|s), p(z_2|s), \ldots, p(z_K|s)), \quad (4)$$

where $p(z_k|s)$ is the conditional probability of topic k given the sentence s. The probability distribution $P(z|s)$ hence tells us how many and which topics this sentence covers[3], and how likely the different topics are for this sentence.

In order to produce a query-focused summary, we also need to represent the query in the latent topic space. This is achieved by folding the query into the trained model. The folding is performed by EM iterations, where the factors $P(w|z)$ are kept fixed, and only the mixing proportions $P(z|q)$ are adapted in each M-step [11]. The representation of sentences and queries in the latent topic space allows us to apply similarity measures in this space. Furthermore, the topic space is much smaller than the original term vector space.

3.2 Computing query-focused and thematic sentence features

Since we are interested in creating a summary that covers the main topics of a document set and is also focused on satisfying a user's information need, specified by a query, we create sentence-level features that attempt to capture these different aspects in the form of per-sentence scores. We then combine the feature scores to arrive at an overall sentence score.

Each of our evaluation data sets contains a title and a narrative for each cluster of topic-related documents. The narrative consists of one or more sentences describing a user's information need. This allows us to compute the following sentence features, where each feature measures the similarity of the sentence's topic distribution S with a "query" topic distribution:

- $r(S, CT)$ - cluster title
- $r(S, N)$ - cluster narrative
- $r(S, T)$ - document title
- $r(S, D)$ - document term vector
- $r(S, C)$ - cluster centroid vector

To compute the features, we fold the title and the narrative of the document clusters, the document titles, and document and cluster term vectors into the trained PLSA model. Query term vectors are preprocessed in the same way as training sentences, except that no sentence splitting is performed. Document and document cluster term vectors are computed by aggregating sentence term vectors.

We evaluate three similarity measures r in our approach: The symmetric Kullback-Leibler (KL) divergence, the Jensen-Shannon (JS) divergence and the cosine similarity, but a variety of other similarity measures can be utilized towards this end.

[3] In terms of topics whose probability is not negligible, i.e. larger than some small quantity ϵ.

The symmetric KL-divergence is defined as follows:

$$\begin{aligned} KL(S,Q) &= D_{KL}(S||Q) + D_{KL}(Q||S) \\ &= \sum_I S(i) \log \frac{S(i)}{Q(i)} \\ &\quad + \sum_I Q(i) \log \frac{Q(i)}{S(i)}. \end{aligned} \quad (5)$$

To use the KL-divergence as a similarity measure, we scale divergence values to $[0, 1]$ and invert by subtracting from 1, hence

$$r_{KL} = 1 - KL(S,Q)_{scaled}. \quad (6)$$

The Jensen-Shannon divergence is a symmetrized and smoothed version of the KL-divergence, computing the KL-divergence of S, Q with respect to the average of the two input distributions. The JS-divergence based similarity r_{JS} is then defined as:

$$\begin{aligned} r_{JS}(S,Q) &= 1 - [D_{JS}(S||Q)] \quad (7) \\ &= 1 - \left[\frac{1}{2} D_{KL}(S||M) + \frac{1}{2} D_{KL}(Q||M)\right], \end{aligned}$$

where $M = 1/2(S + Q)$. Finally, the cosine similarity is defined as $r_{COS}(S, Q) = S^T Q$.

As the training of a PLSA model using the EM algorithm with random initialization converges on a local maximum of the likelihood of the observed data, different initializations will result in different locally optimal models. As the authors of [5] have shown, the effect of random initialization can be reduced by generating several PLSA models, then computing features according to the different models, and finally averaging the feature values. We have implemented this model averaging in our approach using 5 iterations of training the PLSA model.

3.3 Sentence scoring

The system described so far assigns a vector of similarity feature values to each sentence s. The overall score of a sentence s based on the feature vector $(r_1^s, \ldots, r_P^s)$ is:

$$score(s) = \sum_P w_p r_p^s, \quad (8)$$

where w_p is a feature-specific weight. Sentences are ranked by this score, and the highest-scoring sentences are selected for the summary.

For our system, we trained the feature weights by initializing all weights to a default value of 1. We then optimized one feature weight at a time while keeping the others fixed. The training was performed on the DUC 2006 data set. The most dominant features in our experiments are the sentence-narrative similarity $r(S, N)$ and the sentence-document similarity $r(S, D)$, which confirms previous research. On the other hand, the sentence-title similarity $r(S, T)$ did not have a significant influence on the resulting summaries.

When generating a summary, we also need to deal with the problem of repetition of information. This problem is especially important for multi-document summarization, where multiple documents will discuss

System	k	Rouge-1	Rouge-2	Rouge-SU4
PLSA-JS	192	**0.43283**	**0.09698**	**0.15568**
PYTHY	-	-	0.096	0.147
PLSA-COS	256	0.42444	0.09588	0.15409
peer 24	-	0.40980	0.09505	0.15464
PLSA-KL	256	0.42956	0.09465	0.15474
LSI	128	0.42155	0.08880	0.14938
Lead	-	0.30217	0.04947	0.09788

Table 1: *DUC-06: ROUGE recall scores for best number of latent topics k. PLSA-JS, -KL and -COS are system variants using the Jensen-Shannon-divergence, symmetric KL-divergence, and Cosine similarity respectively. Best LSI model based on a rank-k approximation with $k = 128$.*

the same topic. We model redundancy similar to the maximum marginal relevance framework [6]. MMR is a greedy approach that iteratively selects the best-scoring sentence for the summary, and then updates sentence scores by computing a penalty based on the similarity of each sentence with the current summary:

$$score(s) = \lambda(score(s)) - (1 - \lambda)r(S, SUM), \quad (9)$$

where the score of sentence s is scaled to $[0, 1]$ and $r(S, SUM)$ is the cosine similarity of the sentence and the summary centroid vector, which is based on the averaged topic distribution of sentences selected for the summary. λ is set experimentally to 0.5, weighting relevance and redundancy scores equally.

4 Experiments

For the evaluation of our summarization system, we use two data sets from recent summarization tasks: Multi-document summarization in DUC 2006 and in DUC 2007. For all our evaluations, we use ROUGE metrics[4]. ROUGE metrics are recall-oriented and based on n-gram overlap. ROUGE-1 has been shown to correlate well with human judgements [15]. In addition, we also report the performance on ROUGE-2 (bigram overlap) and ROUGE-SU4 (skip bigram) metrics.

We implemented two baseline systems, *Lead* and a system using LSI [9]. The *Lead* system selects the lead sentences from the most recent news article in the document cluster as the summary. The LSI baseline computes the rank-k singular value decomposition of the term-sentence matrix. The resulting right-singular vectors, scaled by the singular values, represent the sentences in the latent semantic space. We compute the same sentence-level features as for the PLSA-based system, using the cosine similarity measure, and apply our greedy ranking and redundancy removal strategy to create a summary.

4.1 DUC 2006

In the multi-document summarization task in DUC-2006, participants are given 50 document clusters, where each cluster contains 25 news articles related to the same topic. Participants are asked to generate summaries of at most 250 words for each cluster. For each cluster, a title and a narrative describing a user's information need are provided. The narrative is usually composed of a set of questions or a multi-sentence task description.

We present the results of our system in Table 1. We compare the results to the best peer (*peer24*) and to the best reported results on this data set by the PYTHY system [25]. In addition, we also give the results for the LSI and the *Lead* baselines.

In the table, system *PLSA-JS* uses the Jensen-Shannon divergence as the similarity measure $r(S, Q)$, *PLSA-KL* the symmetric KL-divergence and *PLSA-COS* the cosine similarity. The results are given for the empirically best value of the parameter k (number of latent topics) for each system variant. The system using the JS-divergence outperforms the best existing systems at $k = 192$ with a ROUGE-2 score of 0.9698, although the improvements for ROUGE-2 and ROUGE-SU4 are not significant at $p < 0.05$. ROUGE-1 scores are significantly better than the results reported by *peer24*. A comparison to the PYTHY system on ROUGE-1 scores was not possible as the authors do not specify this score for their system. All variants of our system outperform the LSI baseline on ROUGE-2.

4.2 DUC 2007

The multi-document summarization task in DUC-2007 is the same as in DUC-2006, with participants asked to produce 250 word multi-document summaries for a total of 45 document clusters. The results of our system are presented in Table 2.

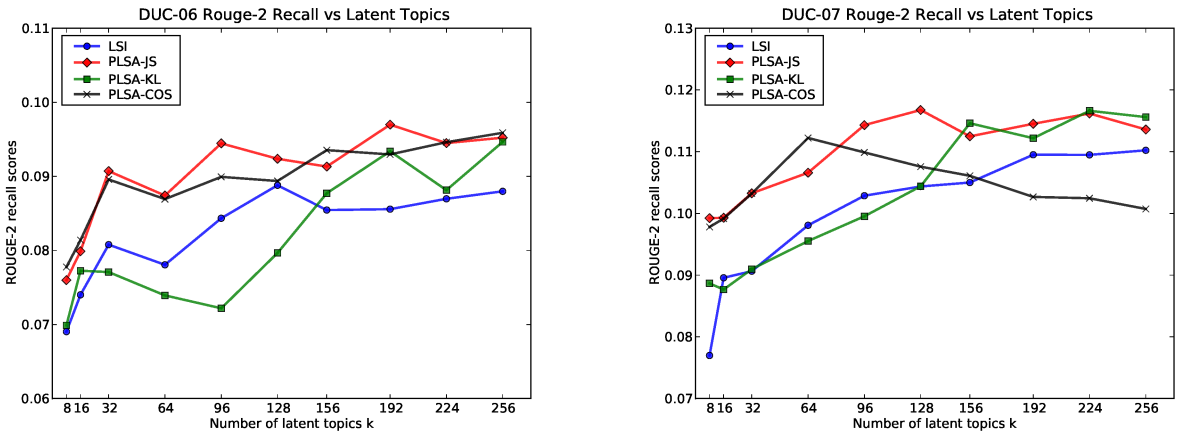
ROUGE-2 and ROUGE-SU4 scores of our system are lower than those of the best system (*peer15*), but still very competitive, with the PLSA-JS variant ranking 5th for ROUGE-2 and 2nd for ROUGE-SU4 when compared to other participating systems. Again we see that all three system variants outperform the LSI baseline. We observe that both the PLSA-JS and the PLSA-COS variant require a much smaller number of latent classes than the LSI model for comparable ROUGE-2 results.

We can also see that the PLSA-JS variant outperforms *peer15* on ROUGE-1, and achieves almost the same score as the top-performing system for ROUGE-SU4, with the differences in both cases not being significant. This suggests that the PLSA model can adequately capture the importance of individual words for ROUGE-1 recall, and word co-occurrences for ROUGE-SU4 skip-bigram recall. The ROUGE-2 score, on the other hand, is significantly lower than that of *peer15*. This indicates that the PLSA model, which was trained on the co-occurrence counts of inviddidual words, could benefit from the inclusion of bigram co-occurrence counts.

4.3 Effect of system variations

Next, we look at the effect of varying the number of latent topics: For all systems we find that using less than $k = 32$ latent classes, the model cannot cope with

[4] ROUGE version 1.5.5, with arguments -n 4 -w 1.2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0



(a) DUC 2006

(b) DUC 2007

Fig. 1: *Summarization performance on DUC 2006 (a) and DUC 2007 data (b) in terms of ROUGE-2 recall for different numbers of latent topics. Also shown are the performance of system variants using different similarity measures and the LSI baseline. The system using Jensen-Shannon divergence with $k = 192$ latent classes outperforms existing approaches on DUC 2006 data.*

System	k	Rouge-1	Rouge-2	Rouge-SU4
peer 15	-	0.44508	0.12448	0.17711
PLSA-JS	128	**0.45843**	**0.11675**	**0.17680**
PLSA-KL	224	0.45208	0.11662	0.17306
PLSA-COS	64	0.44329	0.11222	0.16679
LSI	256	0.44891	0.11022	0.16864
Lead	-	0.31250	0.06039	0.10507

Table 2: *DUC-07: ROUGE recall scores for best number of latent topics k. The PLSA-JS variant outperforms the best participating system on ROUGE-1, and ranks 5th for ROUGE-2.*

the complexity of the data. As can be seen in Figure 1(a), ROUGE-2 scores of the PLSA-JS and -COS variants on DUC 2006 data improve when increasing the number of latent topics k, but level out quickly. From there on, the system seems relatively robust to changes of the number of latent topics, with observed performance variations most likely due to the EM algorithm's convergence on local maxima. The scores of the KL-divergence based variant are significantly lower than those of the PLSA-JS variant when using less than $k = 156$ latent classes, but are almost as good as those of PLSA-JS when using more classes.

Similar observations hold for the DUC 2007 data set, as can be seen in Figure 1(b). One notable difference is that the PLSA-COS variant exhibits overfitting for the DUC 2007 data set much earlier than the other systems, with performance dropping for $k > 64$. This can be explained by the fact that the PLSA model assigns near-zero probabilities to most of the latent classes of a sentence, which in turn affects the cosine similarity more strongly than for the distribution divergence measures which smooth near-zero probabilities.

The most interesting observation for both data sets is that the PLSA-JS variant achieves very good performance with only very few latent classes, signifying that a drastic reduction of the original vector space is possible before computing similarity features. Even with $k = 192$ latent classes, the dimensionality of the space in which we compute similarity features is much lower than for the original term vector space.

Both figures also show that the PLSA approach consistently outperforms the LSI approach. Although performance improvements are not significant at $p < 0.05$, they do indicate that the PLSA model can better capture the sparse information contained in sentences than the LSI model. We find that the LSI baseline system's performance improves significantly when increasing the number of latent classes from 32 to 96, while using more latent classes results in only marginal improvements of ROUGE-2 scores.

5 Related work

Most existing approaches to summarization are sentenced-based and extractive. Proposed approaches explore a variety of features, including document structure and term prominence [18], rhethorical structure [19], as well as graph-theoretic [7, 20] and semantic features [13]. Redundancy is often accounted for by implementing some form of MMR [6], or encoded in feature weighting schemes [25].

Topic information is usually encoded with term-based weighting strategies. Topic signatures [14, 10] are a well-known method of representing topic themes in multi-document summarization. Lexical chains, as proposed by [2], model topic progression through a text using WordNet and other linguistic resources. Different chains can be viewed as modeling different topics within the source documents.

Our system focuses on scoring sentences based on a representation of each sentence in the latent topic space provided by a trained PLSA model. Previous work on applying PLSA to the task of text summarization includes [3], who evaluate single document summarization on DUC 2002 data. The system computes a PLSA model on the term-sentence matrix of a sin-

gle document, then picks the topics with the highest posterior probabilities $p(z)$, and selects sentences with the highest likelihood $p(s|z)$ within these topics for the summary. The approach produces generic summaries based on the most likely topics of the PLSA model. In contrast, our system focuses on query-oriented multi-document summarization, and models redundancy when creating the summary.

More closely related to our approach is recent work by [1], who employ Latent Dirichlet Allocation [4] to create multi-document summaries on DUC 2002 data. The authors report an improvement of ROUGE-1 recall scores over the best known DUC 2002 system. However, their approach is similar to the approach of [3] in being restricted to selecting sentences from the topics with the largest likelihoods. As compared to our approach, their system does not seem to perform any redundancy checking except for relying on the discriminative quality of the latent classes. Furthermore, our approach utilizes narrative and other meta-information of the document cluster to create not only generic, but also query-focused summaries.

6 Conclusion

We introduced an approach to query-focused multi-document summarization based on probabilistic latent semantic analysis. After training a PLSA model on the term-sentence matrix of document clusters from recent summarization tasks, we represent each sentence as a distribution over latent topics. Using this representation, we combine query-focused and thematic sentence features into an overall sentence score. Sentences are ranked and selected for the summary according to this score, choosing a greedy approach for sentence selection and penalizing redundancy with a maximum marginal relevance method.

Our results are among the best reported on the DUC-2006 and DUC-2007 multi-document summarization tasks for ROUGE-1, ROUGE-2 and ROUGE-SU4 scores. Our approach outperforms the previous best performing system on DUC 2006 data, although the improvements are not statistically significant. We have achieved these very competitive results using a simple unsupervised approach. The comparison with a system using latent semantic indexing shows that the PLSA model can better capture the sparse information contained in a sentence than a comparable LSI model. We also studied the effect of different measures to compute sentence-level similarity features in the latent topic space. We found that using the Jensen-Shannon divergence resulted in the best ROUGE scores, as well as being very robust to changes of the number of latent classes.

In future research, we would like to extend our method with additional linguistic knowledge. Given that the unigram, bag-of-words approach ignores syntactic structure information, we would like to study the effect of including such information — by means of bi- or trigram co-occurrence counts — in a PLSA model. The performance differences of our system in terms of ROUGE-2 as compared to ROUGE-1 and ROUGE-SU4 suggests that the model could benefit from including n-gram co-occurrences.

References

[1] R. Arora and B. Ravindran. Latent dirichlet allocation based multi-document summarization. In *Proc. of AND '08*, pages 91–97, 2008.

[2] R. Barzilay and M. Elhadad. *Using Lexical Chains for Text Summarization*, pages 111–121. MIT Press, 1999.

[3] H. Bhandari, M. Shimbo, T. Ito, and Y. Matsumoto. Generic text summarization using probabilistic latent semantic indexing. In *Proc. of IJCNLP 2008*, 2008.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *JMLR*, 3:2003, 2003.

[5] T. Brants, F. Chen, and I. Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proc. of CIKM '02*, pages 211–218, 2002.

[6] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR '98*, pages 335–336, 1998.

[7] G. Erkan and D. Radev. Lexrank: graph-based centrality as salience in text summarisation. *JAIR*, 2004.

[8] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48, 2000.

[9] Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proc. of SIGIR '01*, pages 19–25, 2001.

[10] S. Harabagiu and F. Lacatusu. Topic themes for multi-document summarization. In *Proc. of SIGIR '05*, pages 202–209, 2005.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. of SIGIR '99*, pages 50–57, 1999.

[12] K. S. Jones. Automatic summarising: The state of the art. *Inf. Process. Manage.*, 43(6):1449–1481, 2007.

[13] J. Leskovec, N. Milic-Frayling, and M. Grobelnik. Impact of linguistic analysis on the semantic graph coverage and learning of document extracts. In *Proc. of AAAI'05*, 2005.

[14] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proc. of Coling '00*, pages 495–501, 2000.

[15] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proc. of NAACL-HLT 2003*, pages 71–78, 2003.

[16] C.-Y. Lin and E. Hovy. The potential and limitations of automatic sentence extraction for summarization. In *Proc. of the HLT-NAACL 2003 Workshop on Text Summarization*, pages 73–80, 2003.

[17] H. Luhn. The automatic creation of literature abstracts. *IBM J. of Research & Development*, 1958.

[18] I. Mani and E. Bloedorn. Machine learning of generic and user-focused summarization. In *Proc. of AAAI '98/IAAI '98*, pages 820–826, 1998.

[19] D. Marcu. The rhetorical parsing of natural language texts. In *Proc. of the 35th Annual Meeting of the ACL*, pages 96–103, 1997.

[20] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proc. of ACL 2004*, page 20, 2004.

[21] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proc. of SIGIR '06*, pages 573–580, 2006.

[22] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[23] J. Steinberger, M. A. Kabadjov, M. Poesio, and O. Sanchez-Graillet. Improving LSA-based summarization with anaphora resolution. In *Proc. of HLT-EMNLP '05*, pages 1–8, 2005.

[24] S. Teufel and M. Moens. Sentence extraction as a classification task. In *ACL/EACL-97 Workshop on Intelligent and Scalable Text Summarization*, pages 58–65, 1997.

[25] K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. The PYTHY summarization system: Microsoft research at duc 2007. In *Proc. of DUC 2007*, 2007.