Multi-Document Summarization Using Distortion-Rate Ratio

Ulukbek Attokurov
Department of Computer Engineering
Istanbul Technical University
attokurov@itu.edu.tr

Ulug Bayazit
Department of Computer Engineering
Istanbul Technical University
ulugbayazit@itu.edu.tr

Abstract

The current work adapts the optimal tree pruning algorithm(BFOS) introduced by Breiman et al.(1984) and extended by Chou et al.(1989) to the multi-document summarization task. BFOS algorithm is used to eliminate redundancy which is one of the main issues in multi-document summarization. Hierarchical Agglomerative Clustering algorithm(HAC) is employed to detect the redundancy. The tree designed by HAC algorithm is successively pruned with the optimal tree pruning algorithm to optimize the distortion vs. rate cost of the resultant tree. Rate parameter is defined to be the number of the sentences in the leaves of the tree. Distortion is the sum of the distances between the representative sentence of the cluster at each node and the other sentences in the same cluster. The sentences assigned to the leaves of the resultant tree are included in the summary. The performance of the proposed system assessed with the Rouge-1 metric is seen to be better than the performance of the DUC-2002 winners on DUC-2002 data set.

1 Introduction

Nowadays, the massive amount of information available in the form of digital media over the internet makes us seek effective ways of accessing this information. Textual documents, audio and video materials are uploaded every second. For instance, the number of Google's indexed web pages has exceeded 30 billion web pages in the last two years. Extraction of the needed information from a massive information pool is a challenging task. The task of skimming all the documents in their entirety before deciding which information is relevant is very time consuming.

One of the well known and extensively studied methods for solving this problem is summarization. Text summarization produces a short version of a document that covers the main topics in it (Mani and Hahn, 2000). It enables the reader to determine in a timely manner whether a given document satisfies his/her needs or not.

A single document summarization system produces a summary of only one document whereas a multi-document summarization system produces a summary based on multiple documents on the same topic. Summarization systems can also be categorized as generic or query-based . A generic summary contains general information about particular documents. It includes any information supposed to be important and somehow linked to the topics of the document set. In contrast, a query based summary comprises information relevant to the given query. In this case, query is a rule according to which a summary is to be generated.

Summarization systems can be also classified as extractive or abstractive. In extractive systems, a summary is created by selecting important sentences from a document. Here, only sentences containing information related to the main topics of the document are considered to be important. These sentences are added to the summary without any modification. On the other hand, abstractive systems can modify the existing sentences or even generate new sentences to be included in the summary. Therefore, abstractive summarization is typically more complex than extractive summarization.

The main goal in multi-document summarization is redundancy elimination. Since the documents are related to the same topics, similar text units(passages, sentences etc.) are encountered frequently in different documents. Such text units that indicate the importance of the topics discussed within them should be detected in order to reduce the redundancy. Some of the well-known ap-

Proceedings of the ACL 2014 Student Research Workshop, pages 64–70,
Baltimore, Maryland USA, June 22-27 2014. ©2014 Association for Computational Linguistics

proaches that address this problem are briefly explained in the following section.

Although much work has been done to eliminate the redundancy in multi-document summarization, the problem is still actual and addressed in the current work as well. The current work proposes to integrate the generalized BFOS algorithm (Breiman et al., 1984) adopted by Chou et.al (1989) for pruned tree structured quantizer design with the HAC (Hierarchical Agglomerative Clustering) algorithm. The two main parameters (distortion and rate) in the latter work are adopted to the multi-document summarization task. Distortion can be succinctly defined as the information loss in the meaning of the sentences due to their representation with other sentences. More specifically, in the current context, distortion contribution of a cluster is taken to be the sum of the distances between the vector representations of the sentences in the cluster and representative sentence of that cluster. Rate of a summary is defined to be the number of sentences in the summary, but more precise definitions involving word or character counts are also possible. BFOS based tree pruning algorithm is applied to the tree built with the HAC algorithm. HAC algorithm is used for clustering purposes since BFOS algorithm gets tree structured data as an input. It is found that the suggested approach yields better results in terms of the ROUGE-1 Recall measure (Lin et al., 2003) when compared to 400 word extractive summaries(400E) included in DUC-2002 data set. Also, the results with the proposed method are higher than the ones obtained with the best systems of DUC-2002 in terms of sentence recall and precision(Harabagiu, 2002; Halteren, 2002).

2 Related Works

Term frequency (Luhn, 1958), lexical chains (Barzilay and Elhadad, 1997), location of the sentences (Edmundson, 1969) and the cue phrases (Teufel et al., 1997) are used to determine the important lexical units. Goldstein et al. (2000) proposed a measure named Maximal Marginal Relevance which assigns a high priority to the passages relevant to the query and has minimal similarity to the sentences in the summary. Radev et al. (2001) developed a system called MEAD based on the centroid of the cluster. The words that are most relevant to the main topics are included in the centroid. Lin et al. designed a statistic-based summarization system (Summarist) which incorporated NLP(Natural Language Processing) and IR(Information Retrieval) methods. LSA(Latent Semantic Analysis) (Landauer et al., 1998) has also been used extensively in recent years for multi-document summarization. By applying SVD(Singular Value Decomposition) to the term-document matrix, it determines the most important topics and represents the term and documents in the reduced space (Murray et al., 2005; Steinberger and Jezek , 2004; Geiss, 2011). Rachit Arora et al. (2008) combined LDA(Latent Dirichlet Allocation) and SVD. In this approach, LDA is used to detect topics and SVD is applied to select the sentences representing these topics.

Clustering of the sentences has also been used to determine the redundant information. In this approach, the sentences are first clustered. The sentences in each cluster share common information about the main topics of the documents to be summarized. Then a sentence is selected (Radev et al., 2004) or generated (McKeown et al., 1999) from each cluster that represents the sentences in the cluster. Finally, selected sentences are added to the summary until a predetermined length is exceeded (Aliguliyev, 2006; Hatzivassiloglou et al., 1999; Hatzivassiloglou et al., 2001).

3 Background

3.1 Generalized BFOS Algorithm

Let us assume that we have a tree T with the set of leaves $\tilde{T}$. Also let us denote a sub-tree of T rooted at any node of T as S. The leaves of the sub-trees may happen to be the inner nodes of T. If the root node of the sub-tree S is not identical to the root node of T and the set of leaves $\tilde{S}$ is a sub-set of $\tilde{T}$ then S is called a branch. But if the sub-tree S is rooted at the root node of T then S is named a pruned sub-tree of T. Function defined on the tree T and on any sub-tree S is called a tree functional. Monotonic tree functional is a class of functional where it increases or decreases depending on the tree size. In our case, tree size is the number of the nodes of T.

Two main tree functionals($u1$ and $u2$) need to be defined in the generalized $BFOS$ algorithm. They are adapted to the problem under consideration. In regression trees, $u1$ is the number of the leaves and $u2$ is the mean squared distortion error. In TSVQ(Tree Structured Vector Quantization), $u1$ and $u2$ are the length of the code and the expected distortion, respectively. In the current context, distortion(D) and rate(R) defined in the next section are used as the tree functionals $u1$ and $u2$.

As shown in Chou et al., the set of distortion and rate points of the pruned sub-trees of T generate a convex hull if distortion is an increasing and rate is a decreasing function. Also it is stated that if the tree $T$is pruned off until the root node remains, then it is possible to generate the sub-trees which correspond to the vertices on the lower boundary of the convex hull. Thus it is sufficient to consider the sub-trees corresponding to the vertices of the boundary to trade off between rate and distortion.

A parameter $\lambda = -\frac{\Delta D}{\Delta R}$ may be used to locate the vertices on the lower boundary of the convex hull. ΔD and ΔR indicate the amount of distortion increase and rate decrease when branch subtree S is pruned off. It can be shown that a step on the lower boundary can be taken by pruning off at least one branch sub-tree rooted at a particular inner node. The λ value of this sub-tree is minimal among all the other branch sub-trees rooted at various inner nodes of T, because it is a slope of the lower boundary. At each pruning iteration, the algorithm seeks the branch sub-tree rooted at an inner node with the minimal lambda and prunes it off the tree. After each pruning step, the inner node at which the pruned branch sub-tree is rooted becomes a leaf node. The pruning iterations continue until the root node remains or the pruned sub-tree meets a certain stopping criterion.

4 The Proposed Summarization System

In the current work, BFOS and HAC algorithm were incorporated to the multi-document summarization system. Generalized version of the BFOS algorithm discussed in the work of Chou et al. (1989) with previous applications to TSVQ, speech recognition etc. was adapted for the purpose of pruning the large tree designed by the HAC algorithm. Generalized BFOS algorithm was preferred in the current context because it is believed that the generated optimal trees yield the best trade-off between the semantic distortion and rate (the summary length in terms of number of sentences).

The proposed system consists of the following stages: preprocessing, redundancy detection, redundancy elimination and the summary generation.

In preprocessing stage, the source documents are represented in the vector space. Towards this end, the sentences are parsed, stemmed and a feature set is created (terms (stems or words, n-grams etc.) that occur in more than one document are extracted). The sentences of the document set are then represented by a sentence X term matrix with n columns and m rows, where n is the number of the sentences and m is the number of the terms in the feature set. TF-IDF is used to determine the values of the matrix elements. TF-IDF assigns a value according to the importance of the terms in the collection of the sentences. If the term t occurs frequently in the current document but the opposite is true for other documents then tf-idf value of t is high.

$$TF - IDF = TF * \log \frac{N}{DF} \qquad (1)$$

where TF is the term frequency, DF is the document frequency and N is the number of sentences. Term frequency is the number of the occurrences of the term in the sentence. Document frequency is the number of the sentences in which the term is found.

Redundancy detection is facilitated by applying the Hierarchical Agglomerative Clustering(HAC) algorithm. Initially, individual sentences are considered to be singletons in the HAC algorithm. The most similar clusters are then successively merged to form a new cluster that contains the union of the sentences in the merged clusters. At each step, a new (inner) node is created in the tree as the new cluster appears and contains all the sentences in the union of the merged clusters. HAC merge operations continue until a single cluster remains. The tree built after HAC operation is referred to as the HAC tree.

The third stage is the redundancy elimination. To this end, generalized BFOS algorithm discussed previously is applied to the HAC tree. In order to adapt the generalized BFOS algorithm to the current context, distortion contribution of each cluster (node) is defined as follows:

$$D = \sum_{s \in cluster} d(rs, s) \qquad (2)$$

where d is the distance between the representative sentence(rs) and a sentence(s) in the cluster.

By definition, the distortion contribution of each leaf node of the HAC tree is zero.

Rate is defined to be the number of sentences in the leaves of the tree. A branch sub-tree is removed at each pruning step of the generalized BFOS algorithm. Correspondingly, the sentences at the leaf nodes of the pruned branch subtree are eliminated. As a result, the rate decreases to the number of leaf nodes remaining after pruning.

The centroid of the cluster can be used as the representative sentence of the cluster. Centroid can be constituted of the important (with TF-IDF values exceeding a threshold) words of the cluster (Radev et al., 2004) or can be generated using Natural language processing techniques (McKeown et al., 1999). In the current work, the simpler approach of selecting the sentence from the cluster yielding the minimal distortion as the representative sentence is employed.

λ parameter is used to determine the branch sub-trees that are successively pruned. In each pruning step, the branch sub-tree with minimum λ is identified to minimize the increase in total distortion(ΔD) per discarded sentence(ΔR).

In accordance with the definition of rate given above, ΔR is the change in the number of sentences in the summary before and after the pruning of the branch sub-tree. It also equals to the number of pruned leaf nodes, because rate equals to the number of the sentences stored in the leaf nodes of the current tree. For instance, let us assume that the number of sentences before pruning is 10 and a sub-tree A is cut off. If A has 4 leaf nodes, than 3 of them is eliminated and one is left to represent the cluster of sentences corresponding to the sub-tree A. Since 3 leaf nodes are removed and each leaf node is matched to the certain sentence, the current rate equals to 7. The increase in total distortion is written as

$$\Delta D = D_{post} - D_{prev} \tag{3}$$

where D_{prev} is set equal to the sum of distortions in the leaves of the tree before pruning and D_{post} is set equal to the sum of distortions in the leaves of the tree after pruning.

The application of the generalized BFOS algorithm to the HAC tree can be recapped as follows. At the initial step, a representative sentence is selected for each inner node and λ is determined for each inner node. At each generic pruning step, the node with the minimum lambda value is identified, the sub-tree rooted at that node is pruned, the root node of the sub-tree is converted to a leaf node. After each pruning step, the λ values of the ancestor nodes of this new leaf node are updated. We summarize the generalized BFOS algorithm with a pseudocode in Algorithm 1.

Algorithm 1: PRUNING THE TREE. Prunes a tree T created by using Hierarchical Agglomerative Clustering Algorithm

```
Input: A tree T produced by using
       Hierarchical Clustering Algorithm
Output: Optimal sub-tree $O$ obtained by
        pruning T
For each leaf node,
$\lambda \leftarrow \infty$,$distortion(D) \leftarrow 0$
For each inner node calculate $\lambda = \frac{\Delta D}{\Delta R}$,
where $\Delta D$ and $\Delta R$ are change in
distortion(D) and rate(R) respectively
$rate(R) \leftarrow$ the number of the leaves of T
while the number of the nodes $> 1$ do
    find a node A with minimum $\lambda$ value
    among the inner nodes
    prune the sub-tree S rooted at the node A
    convert the pruned inner node A to the
    leaf node containing the representative
    sentence of the sub-tree S
    update the ancestor nodes of the node A:
    update $\Delta D$, $\Delta R$ and $\lambda$
    update rate(R)
return $O$
```

A summary of desired length can be created by selecting a threshold based on rate (the number of remaining sentences after pruning, the number of leaf nodes of the pruned tree). Another possibility for the choice of the stopping criterion may be based on the λ parameter which monotonically increases with pruning iterations. When a large enough λ value is reached, it may be assumed that shortening the summary further eliminates informative sentences.

The proposed method of summarization has a few drawbacks. The main problem is that the pruning algorithm is highly dependent on the distortion measure. If the distortion measure is not defined appropriately, the representative sentence can be selected incorrectly. Another issue is the inclusion of the irrelevant sentences into the summary. This problem may occur if the sentences remaining after pruning operation are included in the summary without filtering.

5 Evaluation

The testing of the system performed on DUC-2002 data set (Document Understanding Conference, 2002) since the proposed system is designed to produce a generic summary without specified information need of users or predefined user profile. This data set contains 59 document sets. For each document set extraction based summaries with the length 200 and 400 words are provided. Document sets related to the single event are used for testing purposes.

Evaluation of the system is carried out using ROUGE package (Lin C, 2004). Rouge is a summary evaluation approach based on n-gram co-occurrence , longest common subsequence and skip bigram statistics (Lin et al., 2003). The performance of the summarizing system is measured with Rouge-1 Recall, Rouge-1 Precision and F1 measure(Table 1). 400E stood for the extractive 400 word summary provided by DUC-2002 data set. It was created manually as an extractive summary for evaluation purposes. Candidate summary(CS) was produced by the proposed system. Both summaries were compared against a 200 word abstractive summary included in DUC-2002 data set. 200 word abstractive summary was considered as the model summary in ROUGE package. As shown, the summary of the proposed system gives better results in Rouge-1 recall measure. However, the highest precision is achieved in the 400E summary. Generally, the proposed system outperforms the 400E summary, since F1-score which takes into account precision and recall is higher.

In addition, the performance of the system was compared with the best systems(BEST) of DUC-2002(Halteren, 2002; Harabagiu, 2002)(Table 2). The results of the best systems(BEST) in terms of sentence recall and sentence precision are provided by DUC-2002. Sentence recall and sentence precision of the candidate summary(produced by the proposed system) were calculated by using 400 word extract based summary(provided by DUC-2002) and a candidate summary. Sentence recall and sentence precision are defined as follows:

$$sentence\ recall = \frac{M}{B} \tag{4}$$

$$sentence\ precision = \frac{M}{C} \tag{5}$$

where M is the number of the sentences included in both of the summaries(a candidate and 400 word summary provided by DUC-2002(400E)), C,B are the number of the sentences in the candidate summary and in a 400E summary, respectively.

summary	P	R	F1
400E	0.313	0.553	0.382
candidate	0.3	0.573	0.394

Table 1: ROUGE-1 Results. Candidate summary(produced by the proposed system) and 400E summary provided by DUC 2002 are compared with 200 word abstract created manually.

summary	Sentence Precision	Sentence Recall
BEST	0.271	0.272
candidate	0.273	0.305

Table 2: Results. The best systems of DUC-2002 results and the results of the proposed system. Proposed system is compared with 400 word extracts provided by DUC-2002.

As shown, the proposed system performs better than the best systems of DUC-2002 in terms of sentence recall. We are more interested in sentence recall because it states the ratio of the important sentences contained in the candidate summary if the sentences included in the 400E summary are supposed to be important ones. Furthermore, sentence precision is affected from the length of the candidate summary.

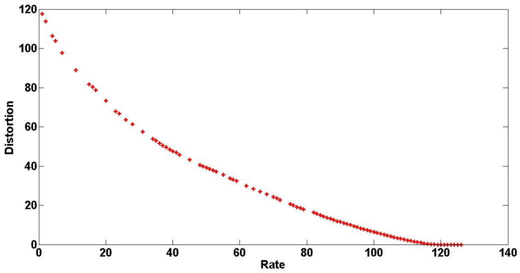

Figure 1: The relationship between distortion and rate. While rate is decreasing distortion is increasing.

Summarizing the text can be considered as the compression of the text. Thus it is possible to depict the graph of dependence of distortion on rate (Figure 1). The graph shows that as rate decreases distortion increases monotonically. Therefore, if distortion is assumed to be the information loss occurred when the original text is summarized then the summaries of different quality can be produced by restricting rate (the number of sentences).

Another graph shows the change of the *lambda* value(Figure 2). The iteration number of the pruning is on X axis and lambda value is on Y one. If λ value of the pruned points are sorted in ascending order and then the graph of ordered λ values is depicted according to their order then the graph identical to the one shown below is obtained(Figure 2). This indicates that the node with minimal lambda value is selected in each iteration. Consequently, the sentences are eliminated so that increase in distortion is minimal for decrease in rate.

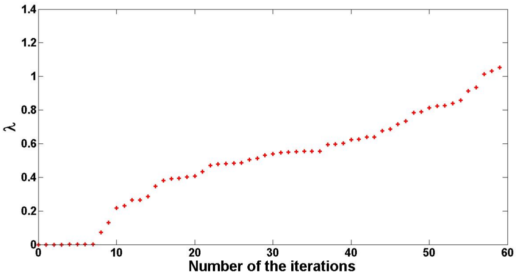

Figure 2: λ value of the pruned node. The change of λ value has upward tendency.

All in all, the quantitative analyses show that the proposed system can be used as one of the redundancy reduction methods. However, in order to achieve the good results, the parameters of BFOS algorithm have to be set appropriately.

6 Conclusion

In this paper , the combination of tree pruning and clustering is explored for the purpose of multi-document summarization. Redundancy in the text detected by the HAC algorithm is eliminated by the generalized BFOS algorithm. It is shown that if the parameters(distortion and rate) are set properly, generalized BFOS algorithm can be used to reduce the redundancy in the text. The depicted graph (Figure 1) shows that the proposed definitions of distortion and rate are eligible for the multi-document summarization purpose.

The performance evaluation results in terms of ROUGE-1 metric suggest that the proposed system can perform better with additional improvements (combining with LSI). Also it is stated that distance measure selection and noisy sentence inclusion have significance impact on the summarization procedure.

Future research will deal with the abstraction. A new sentence will be created(not extracted) when two clusters are merged. It will represent the cluster of sentences as well as summarize the other sentences in the same cluster.

Acknowledgments

We thank Google for travel and conference support for this paper.

References

Aliguliyev R. 2006. A Novel Partitioning-Based Clustering Method and Generic Document Summarization. In *WI-IATW 06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*,pages 626–629,Washington,DC, USA.

Arora R. and Ravindran B. 2008. Latent Dirichlet Allocation Based Multi-Document Summarization. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data (AND 2008)*,91-97.

Barzilay R. and Elhadad M. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*,pages 10-17.

Barzilay R. 2003. Information fusion for multi-document summarization: Paraphrasing and generation, PhD thesis, DigitalCommons@Columbia.

Breiman L., Friedman J.H., Olshen R.A., and Stone C.J. 1984. Classification and Regression Trees. *The Wadsworth Statistics/Probability Series*,Belmont, CA: Wadsworth.

Chou A. Philip, Tom Lookabaugh, and Gray M. Robert. 1989. Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling. *IEEE transactions on information theory*, volume 35, no 2.

DUC–2002. 2002. *Document Understanding Conference.*

Edmundson H. P. 1969. New methods in automatic extracting. *Journal of the ACM*,16:264-285.

Goldstein J., Mittal V., Carbonell J., and Kantrowitz M. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*,pages 40-48.

H. van Halteren. 2002. Writing style recognition and sentence extraction. In *Proceedings of the workshop on automatic summarization*,pages 66–70.

Harabagiu S.M. and Lacatusu F. 2002. Generating single and multi-document summaries with gistexter. In *Proceedings of the workshop on automatic summarization*,pages 30–38.

Hatzivassiloglou V., Klavans J. L., Holcombe M. L., Barzilay R., Kan M.-Y., and McKeown K. R. 1999. Detecting text similarity over short passages: Exploring Linguistic Feature Combinations via Machine Learning. In *Proceedings of the 1999 Joint SIGDAT Conference on empirical Methods in Natural Language Processing and very large corpora*,pages 203-212. College Park, MD, USA.

Hatzivassiloglou V., Klavans J. L., Holcombe M. L., Barzilay R., Kan M.-Y., and McKeown K. R. 2001. SIMFINDER: A Flexible Clustering Tool for Summarization. In *NAACL Workshop on Automatic Summarization*,pages 41-49. Pittsburgh, PA, USA.

Hahn U. and Mani I. 2000. Computer. The challenges of automatic summarization. *IEEE Computer*,33(11),29–36.

Hovy E. and Lin C.Y. 1999. Automated Text Summarization in SUMMARIST. *Mani I and Maybury M (eds.), Advances in Automatic Text Summarization*,pages 81–94. The MIT Press.

Johanna Geiss. 2011. Latent semantic sentence clustering for multi-document summarization, PhD thesis. Cambridge University.

"Towards Multidocument Summarization by Reformulation: Progress and Prospects",

Kathleen McKeown, Judith Klavans, Vasilis Hatzivassiloglou, Regina Barzilay, Eleazar Eskin. 1999. Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *Proceedings of AAAI*,Orlando, Florida.

Landauer T.K., Foltz P.W., and Laham D. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*,25,pages 259–284.

Lin C.Y. and Hovy E. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLTNAACL- 2003)*,pages 71-78.

Lin C–Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*.

Luhn H.P. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*,2(2):159-165.

Murray G., Renals S., and Carletta J. 2005. Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology*.

Radev D. R., Jing H., and Budzikowska M. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies, pages 21-29. In *ANLP/NAACL Workshop on Summarization*, Morristown, NJ, USA.

Radev R., Blair–goldensohn S,Zhang Z. 2001. Experiments in Single and Multi-Docuemtn Summarization using MEAD. In*First Document Under- standing Conference*,New Orleans,LA.

Radev D. R., Jing H., Stys M., and Tam D. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*,40:919-938.

Scott Deerwester, Dumais T. Susan, Furnas W George , Landauer Thomas K., and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*,41(6):391-407.

Steinberger J. and Jezek K. 2004. Using Latent Semantic Analysis in Text Summarization and Summary Evaluation. *Proceedings of ISIM '04*, pages 93-100.

Teufel, Simone, and Marc Moens. 1997. Sentence extraction as a classification task. *ACL/EACL workshop on Intelligent and scalable Text summarization*,58-65.