Combining Optimal Clustering and Hidden Markov Models for Extractive Summarization

Pascale Fung

Human Language Technology Center,
Dept. of Electrical & Electronic Engineering,
University of Science & Technology (HKUST)
Clear Water Bay, Hong Kong
pascale@ee.ust.hk

Grace Ngai

Dept. of Computing,
Hong Kong Polytechnic University,
Kowloon, Hong Kong
csgngai@polyu.edu.hk

CHEUNG, Chi-Shun

Human Language Technology Center,
Dept. of Electrical & Electronic Engineering,
University of Science & Technology (HKUST)
Clear Water Bay, Hong Kong
eepercy@ee.ust.hk

Abstract

We propose Hidden Markov models with unsupervised training for extractive summarization. Extractive summarization selects salient sentences from documents to be included in a summary. Unsupervised clustering combined with heuristics is a popular approach because no annotated data is required. However, conventional clustering methods such as K-means do not take text cohesion into consideration. Probabilistic methods are more rigorous and robust, but they usually require supervised training with annotated data. Our method incorporates unsupervised training with clustering, into a probabilistic framework. Clustering is done by modified K-means (MKM)--a method that yields more optimal clusters than the conventional K-means method. Text cohesion is modeled by the transition probabilities of an HMM, and term distribution is modeled by the emission probabilities. The final decoding process tags sentences in a text with theme class labels. Parameter training is carried out by the segmental K-means (SKM) algorithm. The output of our system can be used to extract salient sentences for summaries, or used for topic detection. Content-based evaluation shows that our method outperforms an existing extractive summarizer by 22.8% in terms of relative similarity, and outperforms a baseline summarizer that selects the top N sentences as salient sentences by 46.3%.

1 Introduction

Multi-document summarization (MDS) is the summarization of a collection of related documents (Mani (1999)). Its application includes the summarization of a news story from different sources where document sources are related by the theme or topic of the story. Another application is the tracking of news stories from the single source over different time frame. In this case, documents are related by topic over time.

Multi-document summarization is also an extension of single document summarization. One of the most robust and domain-independent summarization approaches is extraction-based or shallow summarization (Mani (1999)). In extraction-based summarization, salient sentences are automatically extracted to form a summary directly (Kupiec et. al, (1995), Myaeng & Jang (1999), Jing et. al, (2000), Nomoto & Matsumoto (2001,2002), Zha (2002), Osborne (2002)), or followed by a synthesis stage to generate a more natural summary (McKeown & Radev (1999), Hovy & Lin (1999)). Summarization therefore involves some *theme* or *topic identification* and then extraction of salient segments in a document.

Story segmentation, document and sentence and classification can often be accomplished by unsupervised, clustering methods, with little or no requirement of human labeled data (Deerwester (1991), White & Cardie (2002), Jing et. al (2000)).

Unsupervised methods or hybrids of supervised and unsupervised methods for extractive summarization have been found to yield promising results that are either comparable or superior to supervised methods (Nomoto & Matsumoto (2001,2002)). In these works, vector space models are used and document or sentence vectors are clustered together according to some similarity measure (Deerwester (1991), Dagan et al. (1997)).

The disadvantage of clustering methods lies in their ad hoc nature. Since sentence vectors are considered to be independent sample points, the sentence order information is lost. Various heuristics and revision strategies have been applied to the general sentence selection schema to take into consideration text cohesion (White & Cardie (2002), Mani and Bloedorn (1999), Aone et. al (1999), Zha (2002), Barzilay et al., (2001)). We would like to preserve the natural linear cohesion of sentences in a text as a baseline prior to the application of any revision strategies.

To compensate for the ad hoc nature of vector space models, probabilistic approaches have regained some interests in information retrieval in recent years (Knight & Marcu (2000), Berger & Lafferty (1999), Miller et al., (1999)). These recent probabilistic methods in information retrieval are largely inspired by the success of probabilistic models in machine translation in the early 90s (Brown et. al), and regard information retrieval as a noisy channel problem. Hidden Markov Models proposed by Miller et al. (1999), and have shown to outperform *tf, idf* in TREC information retrieval tasks. The advantage of probabilistic models is that they provide a more rigorous and robust framework to model query-document relations than ad hoc information retrieval. Nevertheless, such probabilistic IR models still require annotated training data.

In this paper, we propose an iterative unsupervised training method for multi-document extractive summarization, combining vectors space model with a probabilistic model. We iteratively classify news articles, then paragraphs within articles, and finally sentences within paragraphs into common story themes, by using modified K-means (MKM) clustering and segmental K-means (SKM) decoding. We obtain an initial clustering of article classes by MKM, which determines the inherent number of theme classes of all news articles. Next, we use SKM to classify paragraphs and then sentences. SKM iterates between a k-means clustering step, and a Viterbi decoding step, to obtain a final classification of sentences into theme classes. Our MKM-SKM paradigm combines vector space clustering model with a probabilistic framework, preserving some of the natural sentence cohesion, without the requirement of annotated data. Our method also avoids any arbitrary or ad hoc setting of parameters.

In section 2, we introduce the modified K-means algorithm as a better alternative than conventional K-means for document clustering. In section 3 we present the stochastic framework of theme classification and sentence extraction. We describe the training algorithm in section 4, where details of the model parameters and Viterbi scoring are presented. Our sentence selection algorithm is described in Section 5. Section 6 describes our evaluation experiments. We discuss the results and conclude in section 7.

2 Story Segmentation using Modified K-means (MKM) Clustering

The first step in multi-document summarization is to segment and classify documents that have a common theme or story. Vector space models can be used to compare documents (Ando et al. (2000), Deerwester et al. (1991)). K-means clustering is commonly used to cluster related document or sentence vectors together. A typical k-means clustering algorithm for summarization is as follows:

1. Arbitrarily choose K vectors as initial centroids;
2. Assign vectors closest to each centroid to its cluster;
3. Update centroid using all vectors assigned to each cluster;
4. Iterate until average intra-cluster distance falls below a threshold;

We have found three problems with the standard k-means algorithm for sentence clustering. First, the initial number of clusters k, has to be set arbitrarily by humans. Second, the initial partition of a cluster is arbitrarily set by thresholding. Hence, the initial set of centroids is arbitrary. Finally, during clustering, the centroids are selected as the sentence among a group of sentences that has the least aver-

age distance to other sentences in the cluster. All these characteristics of K-means can be the cause of a non-optimal cluster configuration at the final stage.

To avoid the above problems, we propose using modified K-means (MKM) clustering algorithm (Wilpon & Rabiner(1985)), coupled with virtual document centroids. MKM starts from a global centroid and splits the clusters top down until the clusters stabilize:

1. Compute the centroid of the entire training set;
2. Assign vectors closest to each centroid to its cluster;
3. Update centroid using all vectors assigned to each cluster;
4. Iterate 2-4 until vectors stop moving between clusters;
5. Stop if clusters stabilizes, and output final clusters, else goto step 6;
6. Split the cluster with largest intra-cluster distance into two by finding the pair of vectors with largest distance in the cluster. Use these two vectors as new centroids, and repeat steps 2-5.

In addition, we do not use any existing document in the collection as the selected centroid. Rather, we introduce virtual centroids that contain the expected value of all documents in a cluster. An element of the centroid is the average weight of the same index term in all documents within that cluster:

$$\vec{\mu}_i = \frac{\sum_{m=1}^{M} w_i^m}{M}$$

The vectors are document vectors in this step. The number of clusters is determined after the clusters are stabilized. The resultant cluster configuration is more optimal and balanced than that from using conventional k-means clustering. Using the MKM algorithm with virtual centroids, we segment the collection of news articles into clusters of related articles. Articles covering the same story from different sources now carry the same theme label. Articles from the same source over different time period also carry the same theme label. In the next stage, we iteratively re-classify each paragraph, and then re-classify each sentence in each paragraph into final theme classes.

3 A Stochastic Process of Theme Classification

After we have obtained story labels of each article, we need to classify the paragraphs and then the sentences according to these labels. Each paragraph in the article is assigned the cluster number of that article, as we assume all paragraphs in the same article share the same story theme.

We suggest that the entire text generation process can be considered as a stochastic process that starts in some *theme* class, generates sentences one after another for that theme class, then goes to the next theme and generates the sentences, so on and so forth, until it reaches the final theme class in a document, and finishes generating sentences in that class. This is an approximation of the authoring process where a writer thinks of a certain structure for his/her article, starts from the first section, writes sentences in that section, proceeds to the next section, etc., until s/he finishes the last sentence in the last section.

Given a document of sentences, the task of summary extraction involves discovering the underlying theme class transitions at the sentence boundaries, classify each sentence according to these theme concepts, and then extract the salient sentences in each theme class cluster.

We want to find $\arg\max_{\vec{C}} P(\vec{C}|\vec{D})$ where $\vec{D}$ is a document consisting of linearly ordered sentence sequences $\vec{D} = (s(1), s(2), \cdots, s(t), \cdots, s(T))$, and $\vec{C}$ is a theme class sequence which consists of the class labels of all the sentences in $\vec{D}$, $\vec{C} = (c(s(1)), c(s(2)), \cdots, c(s(t)), \cdots, c(s(T)))$.

Following Bayes Rule gives us $P(\vec{C}|\vec{D}) = P(\vec{D}|\vec{C})P(\vec{C})/P(\vec{D})$. We assume $P(\vec{D})$ is equally likely for all documents, so that finding the best class sequence becomes:

$$\arg\max_{\vec{C}} P(\vec{C}|\vec{D}) \cong \arg\max_{\vec{C}} P(\vec{D}|\vec{C})P(\vec{C})$$
$$= \arg\max_{\vec{C}} P(s(1), c(s(1)), s(2), c(s(2)), \cdots, s(t), c(s(t)), \cdots s(T), c(s(T))) \cdot$$
$$P(c(s(1)), c(s(2)), \cdots, c(s(t)), \cdots c(s(T)))$$

Note that the total number of theme classes is far fewer than the total number of sentences in a document and the mapping is not one-to-one. Our task is similar to the concept of discourse parsing (Marcu (1997)), where discourse structures are extracted from the text. In our case, we are carrying out discourse *tagging*, whereby we assign the class labels or tags to each sentence in the document.

We use Hidden Markov Model for this stochastic process, where the classes are assumed to be hidden states.

We make the following assumptions:

- The probability of the sentence given its past only depends on its theme class (emission probabilities);
- The probability of the theme class only depends on the theme classes of the previous N sentences (transition probabilities).

The above assumptions lead to a Hidden Markov Model with M states representing M different theme classes. Each state can generate different sentences according to some probability distribution—the emission probabilities. These states are hidden as we only observe the generated sentences in the text. Every theme/state can transit to any other theme/state, or to itself, according to some probabilities—the transition probabilities.

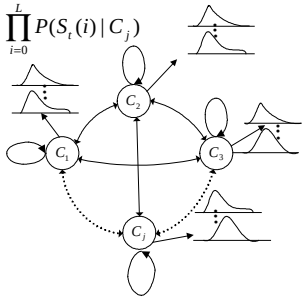

Figure 1: An ergodic HMM for theme tagging

Our theme tagging task then becomes a search problem for HMM: Given the observation sequence $\vec{D} = (s(1), s(2), \cdots, s(t), \cdots, s(T))$, and the model λ , how do we choose a corresponding state sequence $\vec{C} = (c(s(1)), c(s(2)), \cdots, c(s(t)), \cdots, c(s(T)))$, that best explains the sentence sequence?

To train the model parameter λ , we need to solve another problem in HMM: How do we adjust the model parameters $\lambda = (A, B, \pi)$, the transition, emission and initial probabilities, to maximize the likelihood of the observation sentence sequences given our model?

In a supervised training paradigm, we can obtain human labeled class-sentence pairs and carry out a relative frequency count for training the emission and transition probabilities. However, hand labeling some large collection of texts with theme classes is very tedious. One main reason is that there is a considerable amount of disagreement between humans on manual annotation of themes and topics. How many themes should there be? Where should each theme start and end?

It is therefore desirable to decode the hidden theme or topic states using an unsupervised training method without manually annotated data. Consequently, we only need to cluster and label the initial document according to cluster number. In the HMM framework, we then improve upon this initial clustering by iteratively estimate $\lambda = (A, B, \pi)$, and maximize $P(\vec{C} \mid \vec{D})$ using a Viterbi decoder.

3.1 Sentence Feature Vector and Similarity Measure

Prior to the training process, we need to define sentence feature vector and the similarity measure for comparing two sentence vectors.

As we consider a document $\vec{D}$ to be a sequence of sentences, the sentences themselves are represented as feature vectors $s(t)$ of length L, where t is the position of the sentence in the document and L is the size of the vocabulary. Each element of the vector $s(t)$ is an index term in the sentence, weighted by its text frequency (*tf*) and inverse document frequency (*idf*) where *tf* is defined as the frequency of the word in that particular sentence, and *idf* is the inverse frequency of the word in the larger document collection $-\log\frac{df}{N}$ where df is the number of sentences this particular word appears in and N is the total number of sentences in

the training corpus. We select the sentences as documents in computing the *tf* and *idf* because we are comparing sentence against sentence.

In the initial clustering and subsequent Viterbi training process, sentence feature vectors need to be compared to the centroid of each cluster. Various similarity measures and metrics include the cosine measure, Dice coefficient, Jaccard coefficient, inclusion coefficient, Euclidean distance, KL convergence, information radius, etc (Manning & Schütze (1999), Dagan et al. (1997), Salton and McGill (1983)). We chose the cosine similarity measure for its ease in computation:

$$\cos(s(t), s(v)) = \frac{\sum_{i=1}^{L} s(t)_i \cdot s(v)_i}{\sqrt{\sum_{i=1}^{L} (s(t)_i)^2 \cdot \sum_{i=1}^{L} (s(v)_i)^2}}$$

4 Segmental K-means Clustering for Parameter Training

In this section, we describe an iterative training process for estimation of our HMM parameters. We consider the output of the MKM clustering process in Section 2 as an *initial segmentation* of text into class sequences. To improve upon this initial segmentation, we use an iterative Viterbi training method that is similar to the segmental k-means clustering for speech processing (Rabiner & Juang(1993)). All articles in the same story cluster are processed as follows:

1. **Initialization**: All paragraphs in the same story class are clustered again. Then all sentences in the same paragraph shares the same class label as that paragraph. This is the initial class-sentence segmentation. Initial class transitions are counted.
2. **(Re-)clustering**: Sentence vectors with their class labels are repartitioned into K clusters (K is obtained from the MKM step previously) using the K-means algorithm. This step is iterated until the clusters stabilize.
3. **(Re-)estimation of probabilities**: The centroids of each cluster are estimated. Update emission probabilities from the new clusters.
4. **(Re-)classification by decoding**: the updated set of model parameters from step 2 are used to rescore the (unlabeled) training documents into sequences of class given sentences, using Viterbi decoding. Update class transitions from this output.
5. **Iteration**: Stop if convergence conditions are met, else repeat steps 2-4.

The segmental clustering algorithm is iterated until the decoding likelihood converges. The final trained Viterbi decoder is then used to tag unannotated data sets into class-sentence pairs.

In the following Sections 4.1 and 4.2, we discuss in more detail steps 3 and 4.

4.1 Estimation of Probabilities

We need to train the parameters of our HMM such that the model can best describe the training data. During the iterative process, the probabilities are estimated from class-sentence pair sequences either from the initialization stage or the reclassification stage.

4.1.1 Transition Probabilities: Text Cohesion and Text Segmentation

Text cohesion (Halliday and Hasan (1996)) is an important concept in summarization as it underlines the theme of a text segment based on connectivity patterns between sentences (Mani (2002)). When an author writes from theme to theme in a linear text, s/he generates sentences that are tightly linked together within a theme. When s/he proceeds to the next theme, the sentences that are generated next are quite separate from the previous theme of sentences but are they themselves tightly linked again.

As mentioned in the introduction, most extraction-based summarization approaches give certain consideration to the linearity between sentences in a text. For example, Mani (1999) uses spread activation weight between sentence links, (Barzilay et al, 2001) uses a cohesion constraint that led to improvement in summary quality. Anone et al. (1999) uses linguistic knowledge such as aliases, synonyms, and morphological variations to link lexical items together across sentences.

Term distribution has been studied by many NLP researchers. Manning & Schütze (1999) gives a good overview of various probability distributions used to describe how a term appears in a text. The distributions are in general non-Gaussian in nature.

Our Hidden Markov Model provides a unified framework to incorporate text cohesion and term

distribution information in the transition probabilities of theme classes. The class of a sentence depends on the class labels of the previous *N* sentences. The linearity of the text is hence preserved in our model. In the preliminary experiment, we set *N* to be one, that is, we are using a bigram class model.

4.1.2 Emission Probabilities: Poisson distribution of terms

For the emission probabilities, there are a number of possible formulations. We cannot use relative frequency counts of number of sentences in clusters divided by the total sentences in the cluster since most sentences occur only once in the entire corpus. Looking at the sentence feature vector, we take the view that the probability of a sentence vector being generated by a particular cluster is the product of the probabilities of the index terms in the sentence occurring in that cluster according to some distribution, and that these term distribution probabilities are independent of each other.

For a sentence vector of length L, where L is the total size of the vocabulary, its elements—the index terms—have certain probability density function (pdf). In speech processing, spectral features are assumed to follow independent Gaussian distributions. In language processing, several models have been proposed for term distribution, including the Poisson distribution, the two-Poisson model for content and non-content words (Bookstein and Swanson (1975)), the negative binomial (Mosteller and Wallace (1984), Church and Gale (1995)) and Katz's k-mixture (Katz (1996)). We adopt two schemes for comparison (1) the unigram distribution of each index term in the clusters; (2) the Poisson distribution as pdf. for modeling the term emission probabilities:

$$p(k;\lambda_i) = e^{-\lambda_i}\frac{\lambda_i^{\,k}}{k!}$$

At each estimation step of the training process, the λ for the Poisson distribution is estimated from the centroid of each theme cluster. [1]

[1] Strictly speaking, we ought to re-estimate the IDF in the k-mixture during each iteration by using the re-estimated clusters from the k-means step as the documents. However, we simplify the process by using the pre-computed IDF from all training documents.

4.2 Viterbi Decoding: Re-classification with sentence cohesion

After each re-estimation, we use a Viterbi decoder to find the best class sequence given a document containing sentence sequences. The "time sequence" corresponds to the sequence of sentences in a document whereas the states are the theme classes.

At each node of the trellis, the probability of a sentence given any class state is computed from the transition probabilities and the emission probabilities. After Viterbi backtracking, the best class sequence of a document is found and the sentences are relabeled by the class tags.

5 Salient Sentence Extraction

The SKM algorithm is iterated until the decoding likelihood converges. The final trained Viterbi decoder is then used to tag un-annotated data sets into class-sentence pairs. We can then extract salient sentences from each class to be included in a summary, or for question-answering.

To evaluate the effectiveness of our method as a foundation for extractive summarization, we extract sentences from each theme class in each document using four features, namely:

(1) the position of the sentence $p = \frac{1}{\sqrt{n}}$ -- the further it is from the title, the less important it is supposed to be;
(2) the cosine similarity of the sentence with the centroid of its class ψ1;
(3) its similarity with the first sentence in the article ψ2; and
(4) the so-called Z model (Zechner (1996), Nomoto & Matsumoto (2000)), where the *mass* of a sentence is computed as the sum of tf, idf values of index terms in that sentence and the *center of mass* is chosen as the salient sentence to be included in a summary.

$$z = \arg\max_{s} \sum_{i=1}^{L} (1 + \log(\, tf\,(s_i(t)))) \cdot idf\,(s_i(t))$$

The above features are linearly combined to yield a final saliency score for every sentence:

$$w(s) = w_1 \cdot p + w_2 \cdot \psi_1 + w_3 \cdot \psi_2 + w_4 \cdot z$$

Our features are similar to those in an existing system (Radev 2002), with the difference in the centroid computation (and cluster definition), resulting from our stochastic system.

6 Experiments

Many researchers have proposed various evaluation methods for summarization. We find that extrinsic, task-oriented evaluation method to be most easily automated, and quantifiable (Radev 2000). We choose to evaluate our stochastic theme classification system (STCS) on a multi-document summarization task, among other possible tasks We choose a content-based method to evaluate the summaries extracted by our system, compared to those by another extraction-based system MEAD (Radev 2002), and against a baseline system that chooses the top N sentence in each document as salient sentences. All three systems are considered unsupervised.

The evaluation corpus we use is a segment of the English part of HKSAR news from the LDC, consisting of 215 articles. We first use MEAD to extract summaries from 1%-20% compression ratio. We then use our system to extract the *same number* of salient sentences as MEAD, according to the sentence weights. The baseline system also extracts the same amount of data as the other two systems. We plot the cosine similarities of the original 215 documents with each individual extracted summaries from these three systems. The following figure shows a plot of cosine similarity scores against compression ratio of each extracted summary. In terms of relative similarity score, our system is 22.8% higher on average than MEAD, and 46.3% higher on average than the baseline.

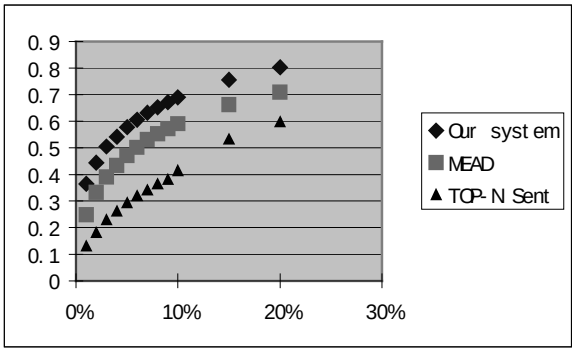

Figure 2: Our system outperforms an existing multi-document summarizer (MEAD) by 22.8% on average, and outperforms the baseline top-N sentence selection system by 46.3% on average.

We would like to note that in our comparative evaluation, it is necessary to normalize all variable factors that might affect the system performance, other than the intrinsic algorithm in each system. For example, we ensure that the sentence segmentation function is identical in all three systems. In addition, index term weights need to be properly trained within their own document clusters. Since MEAD discards all sentences below the length 9, the other two systems also discard such sentences. The feature weights in both our system and MEAD are all set to the default value one. Since all other features are the same between our system and MEAD, the difference in performance is attributed to the core clustering and centroid computation algorithms in both systems.

7 Conclusion and Discussion

We have presented a stochastic HMM framework with modified K-means and segmental K-means algorithms for extractive summarization. Our method uses an unsupervised, probabilistic approach to find class centroids, class sequences and class boundaries in linear, unrestricted texts in order to yield salient sentences and topic segments for summarization and question and answer tasks. We define a class to be a group of connected sentences that corresponds to one or multiple topics in the text. Such topics can be answers to a user query, or simply one concept to be included in the summary. We define a Markov model where the states correspond to the different classes, and the observations are continuous sequences of sentences in a document. Transition probabilities are

the class transitions obtained from a training corpus. Emission probabilities are the probabilities of an observed sentence given a specific class, following a Poisson distribution. Unlike conventional methods where texts are treated as independent sentences to be clustered together, our method incorporates text cohesion information in the class transition probabilities. Unlike other HMM and noisy channel, probabilistic approaches for information retrieval, our method does not require annotated data as it is unsupervised.

We also suggest using modified K-means clustering algorithm to avoid ad hoc choices of initial cluster set as in the conventional K-means algorithm. For unsupervised training, we use a segmental K-means training method to iteratively improve the clusters. Experimental results show that the content-based performance of our system is 22.8% above that of an existing extractive summarization system, and 46.3% above that of simple top-N sentence selection system. Even though the evaluation on the training set is not a close evaluation since the training is unsupervised, we will also evaluate on testing data not included in the training set as our trained decoder can be used to classify sentences in unseen texts. Our framework serves as a foundation for future incorporation of other statistical and linguistic information as vector features, such as part-of-speech tags, name aliases, synonyms, and morphological variations.

References

Chinatsu Aone, James Gorlinsky, Bjornar Larsen, and Mary Ellen Okurowski, 1999. A trainable summarizer with knowledge acquired from robust NLP techniques. In *Advances in automatic text summarization*, ed. Inderjeet Mani and Mark T. Maybury. pp 71-80.

Michele Banko, Vibhu O. Mittal & Michael J. Witbrock. 2000. Headline Generation Based on Statistical Translation. In *Proc. Of the Association for Computational Linguistics*.

Regina Barzilay, Noemie Elhadad & Kathleen R. McKeown. 2001. Sentence Ordering in Multi-document Summarization. In *Proceedings of the 1st Human Language Technology Conference*. pp 149-156. San Diego, CA, US.

Adam Berger & John Lafferty. 1999. Information Retrieval as Statistical Translation. . . In *Proc. Of the 22nd ACM SIGIR Conference (SIGIR-99)*. pp 222-229. Berkeley, CA, USA.

Branimir Boguraev & Christopher Kennedy. 1999. Salience-Based Content Characterisation of Text Documents. In *Advances in automatic text summarization / edited. Inderjeet Mani and Mark T. Maybury*. pp 99-110.

Kenneth Ward Church. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*. pp 136--143. Austin, TX.

Ido Dagan, Lillian Lee, & Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *Proc. Of the 32nd Conference of the Association of Computational Linguistics*, pp 56-63.

Halliday & Hasan, 1976. Cohesion in English. London: Longman.

Marti A. Hearst. 1994. Multi-Paragraph Segmentation of Expository Text. In *Proc. Of the Association for Computational Linguistics*. pp 9-16. Las Cruces, NM.

Eduard Hovy & Chin-Yew Lin. 1999. Automated Text Summarization in SUMMARIST In *Advances in automatic text summarization / edited. Inderjeet Mani and Mark T. Maybury*. pp 81-97.

H. Jing. Dragomir R. Radev and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extaction, utility-based evaluation and user studies. In *Proceedings of ANLP/NAACL-2000.*

Slava M. Katz. 1996. Distribution of content words and phrases in text and language modeling. In *Natural Language Engineering, Vol.2 Part.1*, pp15-60.

Kevin Knight & Daniel Marcu. 2000. Statistics-Based Summarization – Step One: Sentence Compression. In *Proc. Of the 17th Annual Conference of the American Association for Artificial Intelligence*. pp 703-710. Austin, Texas, US.

Julian Kupiec, Jan Pedersen & Francine Chen. 1995. A Trainable Document Summarizer. In *Proc. Of the 18th ACM-SIRGIR Conference.* pp 68-73.

Inderjeet Mani & Eric Bloedorn. 1999. Summarizing Similarities and Differences Among Related Documents. In *Information Retrieval*, 1. pp 35-67.

Christopher D. Manning & Hinrich Schütze 1999. Foundations of statistical natural language processing. *The MIT Press* Cambridge, Massachusetts. London, England.

Daniel Marcu. 1997. The Rhetorical Parsing of Natural Language Texts. In *Proc. of the 35th Annual Meeting of Association for Computational Linguistics and 8th Conference of European Chapter of Association for Computational Linguistics*. pp 96-103. Madrid, Spain.

Bernard Merialdo. 1994. Tagging English Text with a Probabilistic Model. In *Computational Linguistics, 20*.2. pp 155-172.

David R.H. Miller, Tim Leek & Richard M. Schwartz. 1999. A Hidden Markov Model Information Retrieval System. In *Proc. Of the SIGIR'99* pp 214—221. Berkley, CA, US.

Sung Hyon Myaeng & Dong-Hyun Jang. 1999. Development and Evaluation of a Statistically-Based Document Summarization System. In *Advances in automatic text summarization / edited. Inderjeet Mani and Mark T. Maybury*. MIT Press. pp 61-70.

Tadashi Nomoto & Yuji Matsumoto. 2002. Supervised ranking in open domain summarization. In *Proc. Of the 40th Conference of the Association of Computational Linguistics*, pp.. Pennsylvania, US.

Tadashi Nomoto & Yuji Matsumoto. 2001. A New Approach to Unsupervised Text Summarization. In *Proc. Of the SIGIR'01*, pp 26-34 New Orleans, Louisiana, USA.

Jahna C. Otterbacher, Dragomir R. Radev & Airong Luo. 2002. Revision that Improve Cohesion in Multi-document Summaries. In *Proc. Of the Workshop on Automatic Summarization (including DUC 2002)*, pp 27-36. Association for Computational Linguistics. Philadelphia, US.

Miles Osborne. 2002. Using Maximum Entropy for Sentence Extraction. In *Proc. Of the Workshop on Automatic Summarization (including DUC 2002)*, pp 1-8. Association for Computational Linguistics. Philadelphia, US.

Dragomir Radev, Adam Winkel & Michael Topper . 2002. Multi-Document Centroid-based Text Summarization.. In *Proc. Of the ACL-02 Demonstration Session*, pp112-113. Pennsylvania, US.

Michael White & Claire Cardie. 2002. Selecting Sentences for Multidocument Summaries using Randomized Local Search. In *Proc. Of the Workshop on Automatic Summarization (including DUC 2002)*, pp 9-18. Association for Computational Linguistics. Philadelphia, US.

J.G. Wilpon & L.R. Rabiner 1985. A modified K-means clustering algorithm for use in isolated word recognition. In *IEEE Trans. Acoustics, Speech, Signal Proc*. ASSP-33(3), pp 587-594.

K. Zechner. 1996. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *Proc. Of the 16th International Conference on Computational Linguistics*, pp 986-989. Copenhagen, Denmark.

Hongyuan Zha. 2002. Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering. In *Proc. Of the SIGIR'02*. pp 113-120. Tampere, Finland Endre Boros, Paul B. Kantor & David J. Neu. 2001. A Clustering Based Approach to Creating Multi-Document Summaries.