Document Summarization using Conditional Random Fields

Dou Shen[1], **Jian-Tao Sun**[2], **Hua Li**[2], **Qiang Yang**[1], **Zheng Chen**[2]
[1]Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
{dshen, qyang}@cse.ust.hk
[2]Microsoft Research Asia, 49 Zhichun Road, China
{jtsun, huli, zhengc}@microsoft.com

Abstract

Many methods, including supervised and unsupervised algorithms, have been developed for extractive document summarization. Most supervised methods consider the summarization task as a two-class classification problem and classify each sentence individually without leveraging the relationship among sentences. The unsupervised methods use heuristic rules to select the most informative sentences into a summary directly, which are hard to generalize. In this paper, we present a Conditional Random Fields (CRF) based framework to keep the merits of the above two kinds of approaches while avoiding their disadvantages. What is more, the proposed framework can take the outcomes of previous methods as features and seamlessly integrate them. The key idea of our approach is to treat the summarization task as a sequence labeling problem. In this view, each document is a sequence of sentences and the summarization procedure labels the sentences by 1 and 0. The label of a sentence depends on the assignment of labels of others. We compared our proposed approach with eight existing methods on an open benchmark data set. The results show that our approach can improve the performance by more than 7.1% and 12.1% over the best supervised baseline and unsupervised baseline respectively in terms of two popular metrics F_1 and ROUGE-2. Detailed analysis of the improvement is presented as well.

1 Introduction

Document summarization has attracted much attention since the original work by Luhn [Luhn, 1958], which has found wide-ranging applications especially with the explosion of documents on the Internet. Besides its main role of helping readers to catch the main points of a long document with less effort, it is also helpful as a preprocessing step for some text mining tasks such as document classification [Shen *et al.*, 2004].

Document summarization can be categorized along two different dimensions: abstract-based and extract-based. An extract-summary consists of sentences extracted from the document while an abstract-summary may employ words and phrases that do not appear in the original document [Mani, 1999]. The summarization task can also be categorized as either generic or query-oriented. A query-oriented summary presents the information that is most relevant to the given queries, while a generic summary gives an overall sense of the document's content [Goldstein *et al.*, 1999]. In addition to single document summarization, which has been first studied in this field for years, researchers have started to work on multi-document summarization whose goal is to generate a summary from multiple documents that cover similar information. In this paper, we focus on generic single-document sentence extraction which forms the basis for other summarization tasks and is still a hot research topic [Yeh *et al.*, 2005; Mihalcea, 2005].

In the past, extractive summarizers have been mostly based on scoring sentences in the source document based on a set of predefined features [Mani and Bloedorn, 1998]. These features include linguistic features and statistical features, such as location, rhetorical structure [Marcu, 1997], presence or absence of certain syntactic features [Pollock and Zamora, 1975], presence of proper names, statistical measures of term prominence [Luhn, 1958], similarity between sentences, and measures of prominence of certain semantic concepts and relationships [Gong and Liu, 2001]. Two kinds of approaches have been designed to leverage the above features, supervised and unsupervised. In most supervised approaches [Kupiec *et al.*, 1995; Yeh *et al.*, 2005], summarization is seen as a two-class classification problem and the sentences are treated individually. However, we observe that the individual treatment of the sentences cannot take full advantage of the relationship between the sentences. For example, intuitively, two neighboring sentences with similar contents should not be put into a summary together, but when treated individually, this information is lost. Sequential learning systems such as Hidden Markov Models have also been applied, but they cannot fully exploit the rich linguistic features mentioned above since they have to assume independence among the features for tractability [Conroy and O'leary, 2001]. On the other hand, unsupervised approaches rely on heuristic rules that are difficult to generalize. What is ideal for us is to develop a machine learning method based on a training corpus of documents, which can take full advantage of the inter-sentence relationship and rich features which may be dependent.

In this paper, we tackle the extractive summarization problem in a different manner from the above approaches. We take the summarization task as a *sequence labeling problem* instead of a simple classification problem on individual sentences. In our approach, each document is considered as a sequence of sentences and the objective of extractive summarization is to label the sentences in the sequence with 1 and 0, where a label of 1 indicates that the sentence is a summary sentence while 0 denotes a non-summary sentence. The label of one sentence is expected to impact on the labels of other sentences that are nearby. To accomplish this task, we apply conditional random field (CRF) [Lafferty *et al.*, 2001] in this paper, which is a state-of-the-art sequence labeling method. With CRF, we provide a framework for leveraging all the features even if they may be complex, overlapping and not independent. Thus we can fully incorporate our knowledge and intuition of extractive summarization by introducing proper features more effectively. Besides that, the framework can ensemble the outcomes of other summarization methods in a unified way by designing features for them. Our CRF-based approach carries out the summarization task in a discriminative manner, by conditioning the whole label sequence on the sentence sequence, which can maximize the likelihood of the global label sequence as well as maximize the consistency among the different labels in the sequence. As a result, this approach overcomes many of the disadvantages of the previous supervised and unsupervised approaches. The experimental results on an open benchmark data set from DUC01 (http://duc.nist.gov/) show that our proposed approach can improve the performance compared to the state-of-the-art summarization approaches.

2 Related Work

Supervised extractive summarization approaches treat the summarization task as a two-class classification problem at the sentence level, where the summary sentences are positive samples while the non-summary sentences are negative samples. After representing each sentence by a vector of features, the classification function can be trained in two different manners. One is in a discriminative way with well-known algorithms such as Support Vector Machines (SVM) [Yeh *et al.*, 2005]. Although such classifiers are effective, they assume that the sentences are independent and classify each sentence individually without leveraging the relation among the sentences. Hidden Markov Model based methods attempt to break this assumption [Conroy and O'leary, 2001]. In Conroy et al's work, there are two kinds of states, where one kind corresponds to the summary states and the other corresponds to non-summary states. The observations are sentences that are represented by a vector of three features. Given the training data, the state-transition probabilities and the state-specific observation probabilities can be estimated by the Baum-Welch algorithm or an EM algorithm [Rabiner, 1990]. Given a new document, the probability that a sentence corresponds to a summary state can be calculated. Finally, the trained model can be used to select the most likely summary sentences. Although such approaches can handle the positional dependence and feature dependence when the feature space is small by taking some special assumptions, they present two open problems [McCallum *et al.*, 2000]. Firstly, when the feature space is large and the features are not independent or are even overlapping in appearance, the training process will become intractable. Therefore this approach cannot fully exploit the potential useful features that we have mentioned above for the summarization task due to the computational inefficiency. Secondly, the above approaches set the HMM parameters to maximize the likelihood of the observation sequence. By doing so, the approach fails to predict the sequence labels given the observation sequences in many situations because they inappropriately use a generative joint-model in order to solve a discriminative conditional problem when observations are given. Our work in this paper is aimed at solving such problems by CRF.

Many unsupervised methods have been developed for document summarization by exploiting different features and relationships of the sentences as we mentioned above, such as rhetorical structures [Marcu, 1997], lexical chains [Barzilay and Elbadad, 1997], the hidden topics in the documents [Gong and Liu, 2001] and graphs based on the similarity of sentences [Mihalcea, 2005]. The methods based on the last two features require less extra resources and efforts while still achieve better performances compared to other methods, as shown in [Gong and Liu, 2001] and [Mihalcea, 2005]. Therefore, we now review these two works in more detail and compare our approach with them in the experiments.

In [Gong and Liu, 2001], the authors observed that hidden topics can be discovered in a document as well as the projection of each sentence on each topic through Latent Semantic Analysis [Deerwester *et al.*, 1990]. They selected the sentences which have the large projections on the salient topics to form the summary. In Mihalcea's work [Mihalcea, 2005], she constructed a graph in which each node is a sentence and the weight of the edge linking two nodes is the similarity between the corresponding sentences. The direction of the edges can be decided by the appearance order of the sentences. After constructing the graph, she employed some graph-based ranking algorithms like HITS [Kleinberg, 1999] and PageRank [Brin and Page, 1998] to decide the importance of a vertex (sentence) which can take into account the global information recursively computed from the entire graph.

Some previous work has also considered to reduce the redundancy in summary. A typical method is based on the criteria of Maximal Marginal Relevance (MMR) [Carbonell *et al.*, 1997]. According to MMR, a sentence is chosen for inclusion in summary such that it is maximally similar to the document and dissimilar to the already-selected sentences. This approach works in an ad hoc manner and tends to select long sentences. However, in this paper, the redundancy is controlled by a probabilistic model which can be learned automatically.

3 A CRF-based Summarization Approach

3.1 Motivation

Our intuition comes from our observations on how humans summarize a document by posing the problem as a sequence labeling problem. A document can be regarded as a sequence

of sentences that can be partitioned into several segments where each segment is relatively coherent in content. In order to generate a summary with good coverage and low redundancy, we need to select a representative sentence from each segment. Therefore, we have to read the document from the beginning to the end and judge the informativeness of each sentence while reading. If we encounter a sentence which is informative enough, we will put it into the summary. After reading more sentences and encountering better ones, the decision on a previous sentence may be changed. Therefore, the procedure of summarization is kind of sequence labeling. The goal is to produce a label sequence corresponding to the sentence sequence with a label of 1 denoting the summary sentences and 0 denoting the non-summary sentences.

However, the informativeness cannot be easily measured directly by machines. Fortunately, the sentences can be characterized by some features such as their lengths, positions in the article and the terms that they contain. The judgment criteria can be learned from the ground-truth samples generated by people. In other words, given a sequence of sentences represented by certain features, our goal is to label the sentences so that the likelihood of the label sequence given the whole sentence sequence is maximized. In this paper, we use CRF as a tool to model this sequence labeling problem.

3.2 Conditional Random Fields

For a random variable over data sequences to be labeled X, and a random variable over corresponding label sequences Y, Conditional Random Fields (CRF) provide a probabilistic framework for calculating the probability of Y globally conditioned on X [Lafferty *et al.*, 2001]. X and Y may have a natural graph structure. In this paper, we use a common special-case structure, which is a linear chain suitable for sequence labeling. We further assume that there is a one-to-one correspondence between states and labels (two states/labels in our problem: summary sentence and non-summary sentence). Given an observation sequence (sentence sequence here) $X = (x_1, \ldots, x_T)$ and the corresponding state sequence $Y = (y_1, \ldots, y_T)$, the probability of Y conditioned on X defined in CRFs, $P(Y|X)$, is as follows:

$$\frac{1}{Z_X} exp\left(\sum_{i,k} \lambda_k f_k(y_{i-1}, y_i, X) + \sum_{i,l} \mu_l g_l(y_i, X)\right) \quad (1)$$

where Z_X is the normalization constant that makes the probability of all state sequences sum to one; $f_k(y_{i-1}, y_i, X)$ is an arbitrary feature function over the entire observation sequence and the states at positions i and $i-1$ while $g_l(y_i, X)$ is a feature function of state at position i and the observation sequence; λ_k and μ_l are the weights learned for the feature functions f_k and g_l, reflecting the confidence of feature functions. The feature functions can describe any aspect of a transition from y_{i-1} to y_i as well as y_i and the global characteristics of X. For example, f_k may have value 1 when y_{i-1} is a summary sentence while y_i is not a summary sentence and the similarity between x_{i-1} and x_i is larger than a threshold; g_l has a value 1 when y_i is a summary sentence and x_i has upper-case words.

Parameters Estimation

Let $\Lambda = \{\lambda_k, \mu_l\}$ be the set of weights in a CRF model. Λ is usually estimated by a maximum likelihood procedure, that is, by maximizing the conditional log-likelihood of the labeled sequences in the training data $\Psi = \{(X_1, Y_1), \ldots, (X_N, Y_N))\}$, which is defined as:

$$L_\Lambda = \sum_{j=1..N} \log(P_\Lambda(Y_j|X_j)) \quad (2)$$

To avoid overfitting, some regularization methods are employed [Peng and McCallum, 2006]. A common method is to add a Gaussian prior over the parameters:

$$L_\Lambda = \sum_{j=1..N} \log(P_\Lambda(Y_j|X_j)) - \sum_k \frac{\lambda_k^2}{2\sigma_k^2} - \sum_l \frac{\mu_l^2}{2\sigma_l^2} \quad (3)$$

where σ_k^2 and σ_l^2 are the variances of the Gaussian priors.

Various methods can be used to optimize L_Λ, including Iterative Scaling algorithms such as GIS and IIS [Lafferty *et al.*, 2001]. It has been found that a quasi-Newton method such as L-BFGS converges significantly faster [Sha and Pereira, 2003; Malouf, 2002]. Therefore, in this paper, we use L-BFGS.

Inference

Given the conditional probability of the state sequence defined by a CRF in (1) and the parameters Λ, the most probable labeling sequence can be obtained as

$$Y^* = argmax_Y P_\Lambda(Y|X) \quad (4)$$

which can be efficiently calculated with the Viterbi algorithm [Rabiner, 1990]. The marginal probability of states at each position in the sequence can be computed by a dynamic programming inference procedure similar to the forward-backward procedure for HMM [Lafferty *et al.*, 2001]. We can define the "forward values" $\alpha_i(y|X)$ by setting $\alpha_1(y|X)$ equal to the probability of starting with state y and then iterate as follows:

$$\alpha_{i+1}(y|X) = \sum_{y'} \alpha_i(y'|X) exp\left(\Lambda_i(y', y, X)\right) \quad (5)$$

where $\Lambda_i(y', y, X)$ is defined by:

$$\Lambda_i(y', y, X) = \sum_k \lambda_k f_k(y_i = y', y_{i+1} = y, X) + \sum_l \mu_l g_l(y_{i+1} = y, X) \quad (6)$$

Then Z_X equals to $\sum_y \alpha_T(y|X)$. The "backward values" $\beta_i(y|X)$ can be defined similarly. After that, we calculate the marginal probability of each sentence being a summary sentence given the whole sentence sequence by:

$$P(y_i = 1|X) = \frac{\alpha_i(1|X) * \beta_i(1|X)}{Z_X} \quad (7)$$

Thus we can order the sentences based on $P(y_i = 1|X)$ and select the top ones into the summary.

3.3 Feature Space

Many features have been designed for document summarization and can be leveraged through CRF models. In this paper, we use some common features which are widely used in the supervised summarization methods as well as several features induced from the unsupervised methods. The detailed study of other sophisticated features such as the rhetorical relations between sentences is left for future work.

Basic Features

The basic features are the commonly used features in previous summarization approaches, which can be extracted directly without complicated computation [Yeh *et al.*, 2005]. Given a sentence x_i , the features are defined as follows.

Position: the position of x_i along the sentence sequence of a document. If x_i appears at the beginning of the document, the feature "*Pos*" is set to be 1; if it is at the end of the document, "*Pos*" is 2; Otherwise, "*Pos*" is set to be 3.

Length: the number of terms contained in x_i after removing the words according to a stop-word list.

Log Likelihood: the log likelihood of x_i being generated by the document, $logP(x_i|D)$. This is calculated by $\sum_{w_k} N(w_k, x_i) \log p(w_k|D)$ where $N(w_k, x_i)$ is the number of occurrences of w_k in x_i and $p(w_k|D)$ can be estimated by $N(w_k, D)/\sum_{w_j} N(w_j, D)$.

Thematic_Words: these are the most frequent words in the document after the stop words are removed. Sentences containing more thematic words are more likely to be summary sentences. We use this feature to record the number of thematic words in x_i .

Indicator_Words: some words are indicators of summary sentences, such as "in summary" and "in conclusion". This feature is to denote whether x_i contains such words.

Upper_Case_Words: some proper names are often important and presented through upper-case words, as well as some other words the authors want to emphasize. We use this feature to reflect whether x_i contains the upper-case words.

Similarity_to_Neighboring_Sentences: we define features to record the similarity between a sentence and its neighbors. "*Sim_to_Pre_N*" and "*Sim_to_Next_N*" (N = 1, 2, 3) record the similarity of x_i to the previous three sentences and next three sentences respectively. The similarity measurement we use in this work is the cosine similarity.

There are some other popular features such as the number of words in the sentence which are also present in the title, and the position of the sentence in its paragraph. However, since the information about the title and the paragraph is not available in the dataset that we are working on, we do not consider such features in this paper.

Complex Features

LSA_Scores: by decomposing the word-sentence matrix through Singular Vector Decomposition , we can obtain the hidden topics in a document as well as the projection of each sentence on each topic [Gong and Liu, 2001]. Then we can use the projections as scores to rank sentences and select the top sentences into summary. In this paper, we can also treat such projections as features to reflect the importance of the sentences.

HITS_Scores: as shown in the related work section, a document can be treated as a graph and after applying a graph-based ranking algorithm such as HITS or PageRank, each sentence gets a score reflecting its importance. According to [Mihalcea, 2005] and our own experimental results, the authority score of HITS on the directed backward graph is more effective than other graph-based methods. Therefore, we consider only these authority scores and take them as features.

4 Experiments and Results

In this section, we conduct experiments to test our CRF-based summarization approach empirically. The data set is an open benchmark data set which contains 147 document-summary pairs from Document Understanding Conference (DUC) 2001 (http://duc.nist.gov/). We use it because it is for generic single-document extraction task that we are interested in and it is well preprocessed. We denoted it by DUC01.

For the supervised summarization methods, we need to split the data set into training data set and test data set. In order to remove the uncertainty of a data split, a 10-fold cross validation procedure is applied in our experiments, where 9 folds are used for training and one fold for test. Though we do not need to split the data set for unsupervised methods, we apply the unsupervised methods on the same test data as the supervised methods, for the convenience of comparison.

We use two methods to evaluate the results. The first one is by Precision, Recall and F_1 which are widely used in Information Retrieval [Van Rijsbergen, 1979]. For each document, the manually extracted sentences are considered as the reference summary (denoted by S_{ref}). This approach compares the candidate summary (denoted by S_{cand}) with the reference summary and computes the precision, recall and F_1 values as shown in equation (8). We report only F_1 for simplicity, since we come to similar conclusions in our experiments in terms of any of the three measurements.

$$p = \frac{|S_{ref} \bigcap S_{cand}|}{S_{cand}} \; r = \frac{|S_{ref} \bigcap S_{cand}|}{S_{ref}} \; F_1 = \frac{2pr}{p+r} \quad (8)$$

A second evaluation method is by the ROUGE toolkit, which is based on N-gram statistics [Lin and Hovy, 2003]. This tool is adopted by DUC for automatic summarization evaluation that was found to highly correlate with human evaluations. According to [Lin and Hovy, 2003], among the evaluation methods implemented in ROUGE, ROUGE-N (N=1, 2) is relatively simple and works well in most cases. Therefore, we employ only ROUGE-2 for simplicity.

4.1 Baselines

We compare our proposed method with both supervised and unsupervised methods. Among the supervised methods, we choose Support Vector Machine (SVM), Naive Bayes (NB), Logistic Regression (LR) and Hidden Markov Model (HMM). SVM is one of the state-of-the-art classifiers. HMM extends NB by considering the sequential information, while LR is a discriminative version of NB. At the same time, LR can be considered as a linear chain CRF model of order zero and CRF is a discriminative version of HMM. That is, CRF combines the merits of HMM and LR. Recent literature has

claimed the advantages of the discriminative models in classification problems and the effectiveness of sequential information in sequence processing [Sutton and McCallum, 2006]. Therefore, a detailed comparison among these methods can make it clear whether CRF really hold these advantages in our summarization problem.

We also compare our approach with four unsupervised methods. The simplest being to select sentences randomly from the document is denoted as RANDOM. The approach selecting the lead sentences, which is taken as the baseline popularly on the DUC01 dataset, is denoted as LEAD. A similar method is to select the lead sentence in each paragraph. Since the information about the paragraphs is not available in DUC01, we do not include this method as a baseline. Two other unsupervised methods we compare include Gong's algorithm based on LSA and Mihalcea's algorithm based on graph analysis. Among the several options of Mihalcea's algorithm, the method based on the authority score of HITS on the directed backward graph is the best. It is taken by us for comparison. These two unsupervised methods are denoted by LSA and HITS respectively.

4.2 Results and Analysis

Performance based on the basic features

The first experiment compares our CRF-based method with the eight baselines, only using the basic features. Tables 1 and 2 show the results of all the methods in terms of ROUGE-2 and $F1$. We can see that RANDOM is the worst method as expected, while CRF is the best in terms of both evaluation metrics. HITS beats all other baselines, which confirms the effectiveness of graph-based approaches for discovering the importance of sentences. LEAD, by simply selecting the lead sentences, achieves a similar performance to LSA. Both HMM and LR improve the performance as compared to NB due to the advantages of leveraging sequential information and discriminative models. LR and SVM achieve similar performance on the summarization problem. By combining the advantages of HMM and LR together, CRF makes a further improvement by 8.4% and 11.1% over both HMM and LR in terms of ROUGE-2 and F_1, respectively. In fact, CRF is not just a discriminative version of HMM; it is a more powerful method in exploiting dependent features. Due to the same reason, CRF outperforms HITS by 5.3% and 5.7% in terms of ROUGE-2 and F_1, respectively.

	RANDOM	LEAD	LSA	HITS
ROUGE-2	0.245	0.377	0.382	0.431
F_1	0.202	0.311	0.324	0.368

Table 1: Results of unsupervised methods

	NB	LR	SVM	HMM	CRF
ROUGE-2	0.394	0.415	0.416	0.419	0.454
F_1	0.336	0.349	0.343	0.350	0.389

Table 2: Results of supervised methods with basic features

Incorporation of the complex features

The second experiment is to test the effectiveness of the complex features as well as the capability of the supervised methods to incorporate the complex features. The results are shown in Table 3. Compared to the results only based on the basic features, as shown in Table 2, we see that the performance of all the supervised methods are improved significantly. After incorporating the complex features, CRF is still the best method, which improves the values of ROUGE-2 and F_1 achieved by the best baselines by more than 7.1% and 8.8%. Compared with the best unsupervised method HITS, the CRF based on both kinds of features improves the performance by 12.1% and 13.9% in terms of ROUGE-2 and F_1, respectively. In fact, the complex features are the outcomes of the unsupervised methods LSA and HITS. To leverage the complex features through the supervised methods can be thought as a way of combining the outcomes of different methods. In order to test the effectiveness of CRF on combining the outcomes, we compared it to the linear combination method used to combine the results of LSA, HITS and CRF based only on the basic features. By tuning the weight of each method for combination, the best result we can obtain on DUC01 is 0.458 and 0.392 in terms of ROUGE-2 and F_1 respectively, where the improvement is not as significant as CRF based on all the features. Therefore, we can conclude that CRF provides an effective way to combine the outcomes of different methods by treating the outcomes as features.

	NB	LR	SVM	HMM	CRF
ROUGE-2	0.436	0.450	0.449	0.451	0.483
F_1	0.372	0.383	0.385	0.380	0.419

Table 3: Results of supervised methods with all features

	NB	LR	SVM	HMM	CRF
ROUGE-2	0.414	0.427	0.425	0.422	0.470
F_1	0.351	0.365	0.360	0.363	0.411

Table 4: Results of supervised methods with less training data

Effect of the size of the training data

In order to study the impact of the size of the training data on the supervised methods, we conduct a third experiment. We change the training data and test data in the 10-fold cross validation procedure, where one fold is for training and the other nine folds for test. Table 4 shows the results based on both the basic features and the complex features. We can see that the performances of all the supervised methods shown in Table 4 are not as good as those given in Table 3, which is consistent with our intuition, that is we can obtain more precise parameters of the models with more training data. Another observation is that the gap in the performance between CRF-based methods and the other four supervised methods is clearly larger when the size of the training data is small. The reason is that CRF performs better with less training data than HMM since it does not require the features to specify completely a state or observation [Lafferty *et al.*, 2001]. On the other side, HMM, as a generative model, spends a lot of resources on modeling the generative models which are not particularly relevant to the task of inferring the class labels. The bad performance of NB, LR and SVM may be due to the fact that they tend to be overfitting with a small amount of training data.

5 Conclusion and Future Work

In this paper, we have proposed a novel CRF based approach for document summarization, where the summarization task is treated as a sequence labeling problem. By applying the effective sequence labeling algorithm CRF, we provided a framework to consider all available features that include the interactions between sentences. When comparing our CRF-based approach with several existing summarization methods, including the supervised and unsupervised ones on an open data set, we found that our approach can improve the summarization results significantly. The experimental results also validated the capability of our proposed approach to integrate the outcomes of other summarization methods.

In our future work, we plan to exploit more features, especially the linguistic features which are not covered in this paper, such as the rhetorical structures. We will also apply our approach to some more data sets with different genres to test its robustness.

6 Acknowledgements

Dou Shen and Qiang Yang are supported by a grant from NEC (NECLC05/06.EG01). We thank Ms Qionghua Wang and the anonymous reviewers for their useful comments.

References

[Barzilay and Elbadad, 1997] Resina Barzilay and Michael Elbadad. Using lexical chains for text summarization. In *ISTS*, 1997.

[Brin and Page, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[Carbonell *et al.*, 1997] Jaime Carbonell, Yibing Geng, and Jade Goldstein. Automated query-relevant summarization and diversity-based reranking. In *IJCAI-97 Workshop on AI in Digital Libraries*, pages 12–19, Japan, 1997.

[Conroy and O'leary, 2001] John M. Conroy and Dianne P. O'leary. Text summarization via hidden markov models. In *SIGIR*, pages 406–407, 2001.

[Deerwester *et al.*, 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[Goldstein *et al.*, 1999] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: sentence selection and evaluation metrics. In *SIGIR*, pages 121–128, 1999.

[Gong and Liu, 2001] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In *SIGIR*, pages 19–25, 2001.

[Kleinberg, 1999] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5), 1999.

[Kupiec *et al.*, 1995] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *SIGIR*, pages 68–73, 1995.

[Lafferty *et al.*, 2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[Lin and Hovy, 2003] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*, pages 71–78, 2003.

[Luhn, 1958] Hans P. Luhn. The automatic creation of literature abstracts. *IBM J. of R. and D.*, 2(2), 1958.

[Malouf, 2002] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *CoNLL*, 2002.

[Mani and Bloedorn, 1998] Inderjeet Mani and Eric Bloedorn. Machine learning of generic and user-focused summarization. In *AAAI/IAAI*, pages 820–826, 1998.

[Mani, 1999] Inderjeet Mani. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, USA, 1999.

[Marcu, 1997] Daniel Marcu. From discourse structures to text summaries. In *ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, 1997.

[McCallum *et al.*, 2000] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, pages 591–598, 2000.

[Mihalcea, 2005] Rada Mihalcea. Language independent extractive summarization. In *AAAI*, pages 1688–1689, 2005.

[Peng and McCallum, 2006] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *IPM*, 42(4):963–979, 2006.

[Pollock and Zamora, 1975] J.; Pollock and A. Zamora. Automatic abstracting research at chemical abstracts service. *JCICS*, 15(4), 1975.

[Rabiner, 1990] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296, 1990.

[Sha and Pereira, 2003] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *NAACL*, pages 134–141, 2003.

[Shen *et al.*, 2004] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, and Wei-Ying Ma. Web-page classification through summarization. In *SIGIR*, pages 242–249, 2004.

[Sutton and McCallum, 2006] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[Van Rijsbergen, 1979] C. Van Rijsbergen. *Information Retrieval*. 1979.

[Yeh *et al.*, 2005] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng. Text summarization using a trainable summarizer and latent semantic analysis. *IPM*, 41(1):75–95, 2005.