Graph-Based Keyword Extraction for Single-Document Summarization

Marina Litvak
Department of
Information System Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
litvakm@bgu.ac.il

Mark Last
Department of
Information System Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
mlast@bgu.ac.il

Abstract

In this paper, we introduce and compare between two novel approaches, supervised and unsupervised, for identifying the keywords to be used in extractive summarization of text documents. Both our approaches are based on the graph-based syntactic representation of text and web documents, which enhances the traditional vector-space model by taking into account some structural document features. In the supervised approach, we train classification algorithms on a summarized collection of documents with the purpose of inducing a keyword identification model. In the unsupervised approach, we run the HITS algorithm on document graphs under the assumption that the top-ranked nodes should represent the document keywords. Our experiments on a collection of benchmark summaries show that given a set of summarized training documents, the supervised classification provides the highest keyword identification accuracy, while the highest F-measure is reached with a simple degree-based ranking. In addition, it is sufficient to perform only the first iteration of HITS rather than running it to its convergence.

1 Introduction

Document summarization is aimed at all types of electronic documents including HTML files with the purpose of generating the summary - main document information expressed in "a few words".

In this paper, we introduce and compare between two approaches: supervised and unsupervised, for the cross-lingual keyword extraction to be used as the first step in extractive summarization of text documents. Thus, according to our problem statement, the keyword is a word presenting in the document summary.

The supervised learning approach for keywords extraction was first suggested in (Turney, 2000), where parametrized heuristic rules were combined with a genetic algorithm into a system - GenEx - that automatically identified keywords in a document.

For both our approaches, we utilize a graph-based representation for text documents. Such representations may vary from very simple, syntactic ones like words connected by edges representing co-occurrence relation (Mihalcea and Tarau, 2004) to more complex ones like concepts connected by semantic relations (Leskovec et al., 2004). The main advantage of a syntactic representation is its language independency, while the semantic graphs representation provide new characteristics of text such as its captured semantic structure that itself can serve as a document surrogate and provide means for document navigation. Authors of (Leskovec et al., 2004) reduce the problem of summarization to acquiring machine learning models for mapping between the document graph and the graph of a summary. Using deep linguistic analysis, they extract sub-structures (subjectpredicateobject triples) from document semantic graphs in order to get a summary. Contrary to (Leskovec et al., 2004), both our approaches work with a syntactic representation that does not require almost any language-specific linguistic processing. In

Coling 2008: Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, pages 17–24
Manchester, August 2008

this paper, we perform experiments with directed graphs, where the nodes stand for words/phrases and the edges represent syntactic relationships between them, meaning ¨followed by¨ (Schenker et al., 2005).

Some of the most successful approaches to extractive summarization utilize supervised learning algorithms that are trained on collections of "ground truth" summaries built for a relatively large number of documents (Mani and Maybury, 1999). However, in spite of the reasonable performance of such algorithms they cannot be adapted to new languages or domains without training on each new type of data. Our first approach also utilizes classification algorithms, but, thanks to the language-independent graph representation of documents, it can be applied to various languages and domains without any modifications of the graph construction procedure (except for the technical upgrade of implementation for multilingual processing of text, like reading Unicode or language-specific encodings, etc.) (Markov et al., 2007; Last and Markov, 2005). Of course, as a supervised approach it requires high-quality training labeled data.

Our second approach uses a technique that does not require any training data. To extract the summary keywords, we apply a ranking algorithm called HITS (Kleinberg, 1999) to directed graphs representing source documents. Authors of (Mihalcea and Tarau, 2004) applied the PageRank algorithm (Brin and Page, 1998) for keyword extraction using a simpler graph representation (undirected unweighted graphs), and show that their results compare favorably with results on established benchmarks of manually assigned keywords. (Mihalcea and Tarau, 2004) are also using the HITS algorithm for automatic sentence extraction from documents represented by graphs built from sentences connected by similarity relationships. Since we work with directed graphs, HITS is the most appropriate algorithm for our task as it takes into account both in-degree and out-degree of nodes. We show in our experiments that running HITS till convergence is not necessary, and initial weights that we get after the first iteration of algorithm are good enough for rank-based extraction of summary keywords. Another important conclusion that was infered from our experimental results is that, given the training data in the form of annotated syntactic graphs, supervised classification is the most accurate option for identifying the salient nodes in a document graph, while a simple degree-based ranking provides the highest F-measure.

2 Document representation

Currently, we use the "simple" graph representation defined in (Schenker et al., 2005) that holds unlabeled edges representing order-relationship between the the words represented by nodes. The stemming and stopword removal operations of basic text preprocessing are done before graph building. Only a single vertex for each distinct word is created even if it appears more than once in the text. Thus each vertex label in the graph is unique. If a word ***a*** immediately precedes a word ***b*** in the same sentence somewhere in the document, then there is a directed edge from the vertex corresponding to term ***a*** to the vertex corresponding to term ***b***. Sentence terminating punctuation marks (periods, question marks, and exclamation points) are taken by us into account and an edge is not created when these are present between two words. This definition of graph edges is slightly different from co-occurrence relations used in (Mihalcea and Tarau, 2004) for building undirected document graphs, where the order of word occurrence is ignored and the size of the co-occurrence window is varied between 2 and 10. Sections defined for HTML documents are: *title*, which contains the text related to the document's title and any provided keywords (meta-data) and *text*, which comprises any of the readable text in the document. This simple representation can be extended to many different variations like a semantic graph where nodes stand for concepts and edges represent semantic relations between them or a more detailed syntactic graph where edges and nodes are labeled by significant information like frequency, location, similarity, distance, etc. The syntactic graph-based representations were shown in (Schenker et al., 2005) to outperform the classical vector-space model on several clustering and classification tasks. We choose the "simple" representation as a representation that saves processing time and memory resources as well as gives nearly the best results for the two above text mining tasks.

3 Keywords extraction

In this paper, we deal with the first stage of extractive summarization where the most salient words ("keywords") are extracted in order to generate a

summary. Since each distinct word in a text is represented by a node in the document graph, the keywords extraction problem is reduced to the salient nodes extraction in graphs.

3.1 The Supervised approach

In this approach, we try to identify the salient nodes of document graphs by training a classification algorithm on a repository of summarized documents such as (DUC, 2002) with the purpose of inducing a keyword identification model. Each node of every document graph belongs to one of two classes: YES if the corresponding word is included in the document extractive summary and NO otherwise. We consider the graph-based features (e.g., degree) characterizing graph structure as well as statistic-based features (Nobata et al., 2001) characterizing text content represented by a node. The complete list of features, along with their formal definitions, is provided below:

- **In Degree** - number of incoming edges
- **Out Degree** - number of outcoming edges
- **Degree** - total number of edges
- **Frequency** - *term frequency* of word represented by node[1]
- **Frequent words distribution** $\in \{0, 1\}$, equals to 1 iff **Frequency**$\geq$*threshold*[2]
- **Location Score** - calculates an average of location scores between all sentences[3] containing the word N represented by node (denote these sentences as *S(N)*):

$$Score\left(N\right) = \frac{\sum_{S_i \in S(N)} Score\left(S_i\right)}{\left|S\left(N\right)\right|}$$

- **Tfidf Score** - calculates the *tf-idf* score (Salton, 1975) of the word represented by node[4].
- **Headline Score** $\in \{0, 1\}$, equals to 1 iff document headline contains word represented by node.

3.2 The Unsupervised approach

Ranking algorithms, such as Kleinberg's HITS algorithm (Kleinberg, 1999) or Google's PageRank (Brin and Page, 1998) have been elaborated and used in Web-link analysis for the purpose of optimizating the search performance on the Web. These algorithms recursively assign a numerical weight to each element of a hyperlinked set of documents, determining how important each page is. A hyperlink to a page counts as a vote of support. A page that is linked to by many important pages (with high rank) receives a high rank itself. A similar idea can be applied to lexical or semantic graphs extracted from text documents, in order to extract the most significant blocks (words, phrases, sentences, etc.) for the summary (Mihalcea and Tarau, 2004; Mihalcea, 2004). In this paper, we apply the HITS algorithm to document graphs and evaluate its performance on automatic unsupervised text unit extraction in the context of the text summarization task. The HITS algorithm distinguishes between "authorities" (pages with a large number of incoming links) and "hubs" (pages with a large number of outgoing links). For each node, HITS produces two sets of scores - an "authority" score, and a "hub" score:

$$HITS_A\left(V_i\right) = \sum_{V_j \in In(V_i)} HITS_H\left(V_j\right) \quad (1)$$

$$HITS_H\left(V_i\right) = \sum_{V_j \in Out(V_i)} HITS_A\left(V_j\right) \quad (2)$$

For the total rank (H) calculation we used the following four functions:

1. rank equals to the authority score
$$H\left(V_i\right) = HITS_A\left(V_i\right)$$
2. rank equals to the hub score
$$H\left(V_i\right) = HITS_H\left(V_i\right)$$
3. rank equals to the average between two scores
$$H\left(V_i\right) = avg\left\{HITS_A\left(V_i\right), HITS_H\left(V_i\right)\right\}$$
4. rank equals to the maximum between two scores
$$H\left(V_i\right) = \max\left\{HITS_A\left(V_i\right), HITS_H\left(V_i\right)\right\}$$

[1] The term frequency (TF) is the number of times the word appears in a document divided by the number of total words in the document.

[2] In our experiment the threshold is set to 0.05

[3] There are many variants for calculating sentence location score (Nobata et al., 2001). In this paper, we calculate it as an reciprocal of the sentence location in text: $Score\left(S_i\right) = \frac{1}{i}$

[4] There are many different formulas used to calculate *tfidf*. We use the next formula: $\frac{tf}{tf+1} \log_2 \frac{|D|}{df}$, where tf - term frequency (as defined above), $|D|$ - total number of documents in the corpus, *df* - number of documents where the term appears.

average merit	rank	feature
0.192 +- 0.005	1	Frequent words distribution
0.029 +- 0	2	In Degree
0.029 +- 0	3	Out Degree
0.025 +- 0	4	Frequency
0.025 +- 0	5	Degree
0.017 +- 0	6	Headline Score
0.015 +- 0	7	Location Score
0.015 +- 0.001	8	Tfidf Score

Table 1: Feature selection results according to GainRatio value

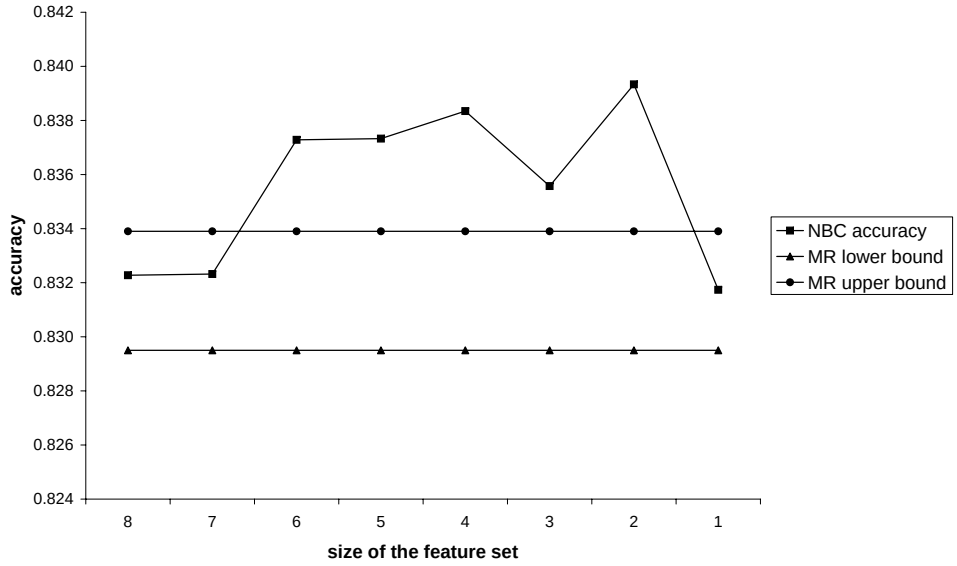

Figure 1: Accuracy for NaïveBayes classifier (NBC) and Majority Rule (MR)

4 Experimental results

All experiments have been performed on the collection of summarized news articles provided by the Document Understanding Conference 2002 (DUC, 2002). This collection contains 566 English texts along with 2-3 summaries per document on average. The size[5] of syntactic graphs extracted from these texts is 196 on average, varying from 62 to 876.

4.1 Supervised approach

We utilized several classification algorithms implemented in Weka's software (Witten and Frank, 2005) : J48 (known as C4.5), SMO (Support Vector Machine) and NaïveBayes for building binary classification models (a word belongs to summary / does not belong to the summary). For the training we built dataset with two classes: YES for nodes belonging to at least one summary of the document, and NO for those that do not belong to any summary. The accuracy of the default (majority) rule over all nodes is equal to the percentage of non-salient nodes (83.17%). For better classification results we examined the importance of each one of the features, described in Section 3.1 using automated feature selection. Table 1 presents the average GainRatio[6] values ("merits") and the average rank of the features calculated from the DUC 2002 document collection, based on 10-fold cross validation.

As expected, the results of J48 and SMO (these algorithms perform feature selection while building the model) did not vary on different feature sets, while NaïveBayes gave the best accuracy on the reduced set. Figure 1 demonstrates the accuracy variations of NaïveBayes classifier on the different feature sets relative to the confidence inter-

[5] We define the size of a graph as the number of its vertices.

[6] $Gain\_Ratio(A) = \frac{Information\_Gain(A)}{Intrinsic\_Info(A)}$, where $Intrinsic\_Info(A) = -\sum_x \frac{N_x}{N} \log\left[\frac{N_x}{N}\right]$

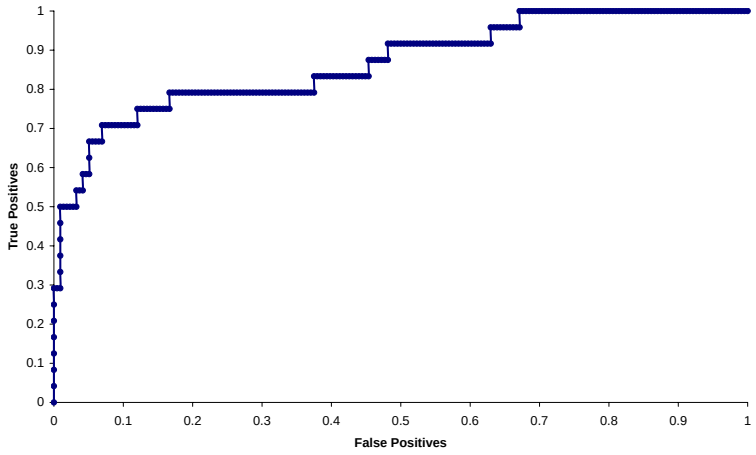

Figure 2: Sample ROC curve for one of the DUC'02 documents

Ranking function	Degree vectors	Converged vectors
Authority	0.625	0.600
Hub	0.620	0.601
Avg(Authority, Hub)	0.651	0.622
Max(Authority, Hub)	0.651	0.624

Table 2: Average AUC for each rank calculating function

val for the majority rule accuracy according to the normal approximation of the binomial distribution with $\alpha = 0.05$. Table 3 presents classification results for supervised algorithms (for NaïveBayes the results shown on the top 2 features) based on 10-fold cross validation as well as results of unsupervised learning.

4.2 Unsupervised approach

We have studied the following research questions:

1. Is it possible to induce some classification model based on HITS scores?
2. Is it necessary to run HITS until convergence?

In order to answer these questions we performed the following two experiments:

1. In the first one, we run HITS only one iteration. Note, that the ranks resulted from the first iteration are just in-degree and out-degree scores for each node in graph, and may be easily computed without even starting HITS[7].
2. In the second experiment we run HITS until convergence[8] (different number of steps for different graphs) and compare the results with the results of the first experiment.

After each experiment we sorted the nodes of each graph by rank for each function (see the rank calculating functions described in Section 3.2). After the sorting we built an ROC (Receiver Operating Characteristic) curve for each one of the graphs. Figure 2 demonstrates a sample ROC curve for one of the documents from DUC 2002 collection.

In order to compare between ranking functions (see Section 3.2) we calculated the average of AUC (Area Under Curve) for the 566 ROC curves for each function. Table 2 presents the average AUC results for the four functions. According to these results, functions that take into account both scores (average and maximum between two scores) are optimal. We use the *average* function for comparing and reporting the following results. Also, we can see that degree vectors give better AUC results

[7]Initially, both authority and hub vectors (a and h respectively) are set to $u = (1, 1, \ldots, 1)$. At each iteration HITS sets an authority vector to $a = A^T h$, and the hub vector to $h = Aa$, where A is an adjacency matrix of a graph. So, after the first iteration, $a = A^T u$ and $h = Au$, that are the vectors containing in-degree and out-degree scores for nodes in a graph respectively.

[8]There are many techniques to evaluate the convergence achievement. We say that convergence is achieved when for any vertex i in the graph the difference between the scores computed at two successive iterations falls below a given threshold: $\frac{|x_i^{k+1} - x_i^k|}{x_i^k} < 10^{-3}$ (Kamvar, 2003; Mihalcea and Tarau, 2004)

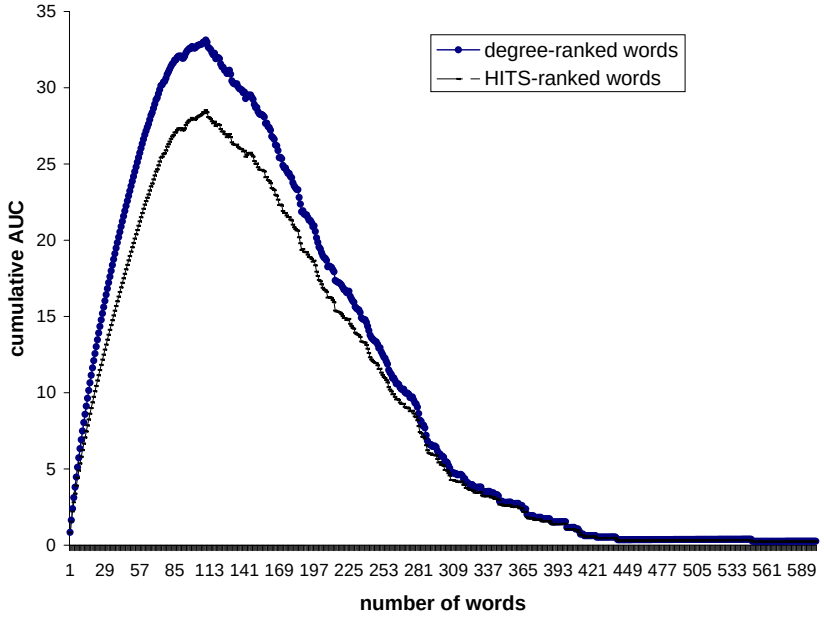

Figure 3: Cumulative AUC curves for degree and converged vectors

Method		Accuracy	TP	FP	Precision	Recall	F-Measure
Classification	J48	**0.847**	0.203	0.022	0.648	0.203	0.309
	NaïveBayes	0.839	0.099	0.011	0.648	0.099	0.172
	SMO	0.839	0.053	**0.002**	**0.867**	0.053	0.100
Degree-based Ranking	$N = 10$	0.813	0.186	0.031	0.602	0.186	0.282
	$N = 20$	0.799	0.296	0.080	0.480	0.296	0.362
	$N = 30$	0.772	0.377	0.138	0.409	0.377	0.388
	$N = 40$	0.739	**0.440**	0.200	0.360	**0.440**	**0.392**

Table 3: Results for each supervised and unsupervised method

than converged ones.

In order to compare between the degree-based vectors and the converged ones we calculated the precision curves[9] for each graph in both experiments. Then for each ranking method the curve representing an average cumulative AUC over the 566 precision curves was calculated. Figure 3 demonstrates the difference between resulting curves. As we can conclude from this chart, the degree-based vectors have a slight advantage over the converged ones. The "optimum" point where the average AUC is maximum for both methods is 111 words with the average AUC of 28.4 for degree-based words and 33 for HITS-ranked words. That does not have much significance because each document has a different "optimum" point.

[9]For each number of top ranked words the percentage of positive words (belonging to summary) is shown.

Finally, we compared the results of unsupervised method against the supervised one. For this purpose, we consider unsupervised model based on extracting top N ranked words for four different values of N: 10, 20, 30 and 40. Table 3 represents the values for such commonly used metrics as: Accuracy, True Positive Rate, False Positive Rate, Precision, Recall and F-Measure respectively for each one of the tested methods. The optimal values are signed in bold.

Despite the relatively poor accuracy performance of both approaches, the precision and recall results for the unsupervised methods show that the classification model, where we choose the top most ranked words, definitely succeeds compared to the similar keyword extraction methods. (Leskovec et al., 2004) that is about "logical triples" extraction rather than single keyword extraction, presents results on DUC 2002 data, which are similar to ours in terms of the F-measure (40%

against 39%) though our method requires much less linguistic pre-processing and uses a much smaller feature set (466 features against 8). (Mihalcea and Tarau, 2004) includes a more similar task to ours (single keyword extraction) though the definition of a keyword is different ("keywords manually assigned by the indexers" against the "summary keywords") and a different dataset (Inspec) was used for results presentation.

5 Conclusions

In this paper we have proposed and evaluated two graph-based approaches: supervised and unsupervised, for the cross-lingual keyword extraction to be used in extractive summarization of text documents. The empirical results suggest the following. When a large labeled training set of summarized documents is available, the supervised classification is the most accurate option for identifying the salient keywords in a document graph. When there is no high-quality training set of significant size, it is recommended to use the unsupervised method based on the node degree ranking, which also provides a higher F-measure than the supervised approach. The intuition behind this conclusion is very simple: most words that are highly "interconnected" with other words in text (except stop-words) should contribute to the summary. According to our experimental results, we can extract up to 15 words with an average precision above 50%. Running HITS to its convergence is redundant, since it does not improve the initial results of the degree ranking.

6 Future work

The next stage of our extractive summarization methodology is generation of larger units from the selected keywords. At each step, we are going to reduce document graphs to contain larger units (subgraphs) as nodes and apply some ranking algorithms to the reduced graphs. This algorithm is iterative, where graph reduction steps are repeated until maximal subgraph size is exceeded or another constraint is met. Also, we plan to work on the supervised classification of sub-graphs, where many graph-based features will be extracted and evaluated.

In the future, we also intend to evaluate our method on additional graph representations of documents, especially on the concept-based representation where the graphs are built from the concepts fused from the texts. Once completed, the graph-based summarization methodology will be compared to previously developed state-of-the-art summarization methods and tools. All experiments will include collections of English and non-English documents to demonstrate the cross-linguality of our approach.

References

S. Brin and L. Page. 1998. *The anatomy of a large-scale hypertextual Web search engine*. *Computer Networks and ISDN Systems*, 30:1–7.

Document Understanding Documents 2002 [http://www-nlpir.nist.gov/projects/duc/index.html]

Sepandar D. Kamvar, Taher H. Haveliwala, and Gene H. Golub. *Adaptive methods for the computation of pagerank*. Technical report, Stanford University.

Kleinberg, J.M. 1999. *Authoritative sources in a hyperlinked environment*. *Journal of the ACM*, 46(5):604-632.

Last, M. and Markov A. 2005. *Identification of terrorist web sites with cross-lingual classiffication tools*. In Last, M. and Kandel, A. (Editors), Fighting Terror in Cyberspace. *World Scientific, Series in Machine Perception and Artificial Intelligence*, 65:117–143.

Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. *Learning Semantic Graph Mapping for Document Summarization*. In Proceedings of ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies.

Mani, I. and Maybury, M.T. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

Markov A., Last, M. and Kandel, A. 2007. *Fast Categorization of Web Documents Represented by Graphs*. Advances in Web Mining and Web Usage Analysis - 8th International Workshop on Knowledge Discovery on the Web, WEBKDD 2006, Revised Papers, O. Nasraoui, et al. (Eds). *Springer Lecture Notes in Computer Science* 4811:56–71.

Mihalcea R. 2004. *Graph-based ranking algorithms for sentence extraction, applied to text summarization*. In Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics, Barcelona, Spain.

Mihalcea and P. Tarau. 2004. *TextRank - bringing order into texts*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.

Martin F. Porter. 1980. *An algorithm for suffix stripping*. *Program*, 14(3):130137, July.

Nobata, C., Sekine, S., Murata, M., Uchimoto, K., Utiyama, M. and Isahara, H. 2001. *Sentence extraction system assembling multiple evidence*. In Proceedings of the Second NTCIR Workshop Meeting, 5–213–218.

Salton, G., Wong, A. and Yang, C. S. 1975. *A Vector Space Model for Automatic Indexing Communications of the ACM*, 18(11):613-620.

Schenker, A., Bunke, H., Last, M., Kandel, A. 2005. *Graph-Theoretic Techniques for Web Content Mining*, volume 62. World Scientific, Series in Machine Perception and Artificial Intelligence.

Peter D. Turney. 2000. *Learning Algorithms for Keyphrase Extraction*. *Information Retrieval*, 2(4):303–336.

Ian H. Witten and Eibe Frank 2005. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.