SIMFINDER: A Flexible Clustering Tool for Summarization

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe,
Regina Barzilay, **Min-Yen Kan**, and **Kathleen R. McKeown**

Department of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027, USA

{vh, klavans, mlh40, regina, min, kathy}@cs.columbia.edu

Abstract

We present a statistical similarity measuring and clustering tool, SIMFINDER, that organizes small pieces of text from one or multiple documents into tight clusters. By placing highly related text units in the same cluster, SIMFINDER enables a subsequent content selection/generation component to reduce each cluster to a single sentence, either by extraction or by reformulation. We report on improvements in the similarity and clustering components of SIMFINDER, including a quantitative evaluation, and establish the generality of the approach by interfacing SIMFINDER to two very different summarization systems.

1 Introduction

Summarization is an application that cuts across multiple natural language processing areas (search, text analysis, planning, generation) and for which disparate approaches have been used, including word counts (Luhn, 1958), information retrieval based similarity measures (Salton et al., 1997), statistical models (Kupiec et al., 1995), positional information (Lin and Hovy, 1997), and discourse structure (Marcu, 1997). For multidocument summarization, where the source texts often contain the same information with variations in the presentation, an alternative approach is to explicitly seek similar pieces of the input text, on the assumption that recurring text units are probably the more central ones. Each set of similar text pieces can then produce one sentence in the summary, either by extraction or by reformulation.

In this paper, we present a statistical similarity and clustering tool that accomplishes the task of finding similar text units (sentences or paragraphs) for summarization, a task that is a component of

multiple current multidocument summarization systems (Mani and Bloedorn, 1997; Carbonell and Goldstein, 1998; McKeown et al., 1999; Radev et al., 2000). Our tool, SIMFINDER, incorporates linguistic features and a sophisticated clustering algorithm to construct sets of highly similar sentences or paragraphs for summarization. In earlier work (Hatzivassiloglou et al., 1999), we discussed how SIMFINDER's text features were selected and evaluated; we summarize these results in Section 2, along with recent improvements on feature selection and weighting. In Section 3, we discuss the clustering algorithm we adopted and modifications to it specific to the summarization task. Finally, in Section 4 we demonstrate the flexibility and generality of the approach by showing how clusters of paragraphs produced by SIMFINDER are used in two summarization systems at our institution, MULTI-GEN and CENTRIFUSER. We present three implemented techniques for the summary generation task, and outline several other possibilities.

Our focus in the present paper is on describing the incremental but significant improvements in SIMFINDER's features, machine learning model, and clustering algorithm over the earlier 1999 version (Hatzivassiloglou et al., 1999); and on offering evidence of the approach's generality by showing how SIMFINDER was successfully interfaced with different (both extraction-based and reformulation-based) content selection and presentation systems for summarization.

2 Using Machine Learning to Compute Similarities

Clustering entails both developing a similarity metric and choosing an appropriate clustering algorithm. In clustering documents for information re-

U.N. Human Rights Commissioner Mary **Robinson** made a landmark visit to **Mexico** at the **government**'s **invitation** after voicing alarm **last year** of **violence** in the **country**'s conflict-torn southern state of **Chiapas**.

Mexico's **government last year** rejected suggestions the United Nations might mediate in the long-running **Chiapas** conflict, saying it could solve its own internal affairs. But it did **invite Robinson** and a special rapporteur on extrajudicial **killings** to come and assess human rights for themselves in the **country**.

Figure 1: Two similar paragraphs; in bold, we highlight the primitive features indicating similarity that are captured by SIMFINDER.

An OH-58 helicopter, carrying a crew of two, was on a routing training orientation when **contact** was **lost** at about 11:30 a.m. Saturday (9:30 p.m. EST Friday).

"There were two people on board," said Bacon. "We **lost** radar **contact** with the helicopter about 9:15 EST (0215 GMT).

Figure 2: A composite feature over word primitives, with the restriction that one primitive must be a noun and one must be a verb.

trieval purposes, as in the recent Topic Detection and Tracking (TDT) efforts (Allan et al., 1998; Fiscus et al., 1999), the similarity measure is usually based on shared words only. This is often appropriate for classification of documents into topics, although even for document-level clustering, the use of linguistically informed features such as named entity tags can improve performance (Hatzivassiloglou et al., 2000). However, we have found that more specialized information can be utilized when we have to work with smaller units of text (sentences or paragraphs) and we want to put together only very similar units, as is the case with summarization. In fact, SIMFINDER is designed to handle input that has already been organized into groups of documents tightly connected on topic and date, either with a separate TDT-like clustering tool or because the input naturally comes in that form.

In our first presentation of SIMFINDER's approach to summarization (Hatzivassiloglou et al., 1999), we identified 43 features that we could efficiently extract from the text and that could plausibly help determine the semantic similarity of two short text units. We chose to use paragraphs, rather than sentences, as our unit of text in most experiments because a paragraph is more likely to contain background information (such as proper nouns) relevant to semantic comparison. Paragraphs in news documents often consist of a single sentence in any case. Figure 1 illustrates some of these features by means of two example similar paragraphs from our training corpus.

These paragraphs have quite a few words in common, including *government*, *last*, *year*, and *country*. Perhaps more significantly, they share several proper nouns: *Robinson*, *Mexico*, and *Chiapas*, which perhaps should be weighted more for a match. Other similarities include words with the same stem, such as *invitation* and *invite*, and semantically related words such as *killings* and *violence*. In all, our set of *primitive* features includes several ways to define a match on a given word: we consider matches involving identical words, as well as words that matched on their stem, as noun phrase heads ignoring modifiers, and as WordNet (Miller et al., 1990) synonyms. These matches of primitive features are further constrained by part of speech and combined to form *composite* features attempting to capture syntactic patterns where two primitive features have to match within a window of five words (not including stopwords). The composite features approximate in this manner syntactic relationships such as subject-verb or verb-object (see Figure 2). In other cases, a composite feature can serve as a more effective version of a single primitive feature. For example, Figure 3 illustrates a composite feature involving WordNet primitives (i.e., words match if they share immediate hypernyms in WordNet) and exact word match primitives. On its own, the WordNet feature might introduce too much noise, but in conjunction with the exact word match feature it can be a useful indicator of similarity.

For the purpose of automatic feature selection, we developed a data set consisting of 10,535 manually marked pairs of paragraphs from the Reuters part of the 1997 TDT pilot corpus. Each pair of paragraphs was judged by two human reviewers, working separately. The reviewers were asked to make a binary determination on whether the two

Boris Yeltsin was hospitalized Monday with what doctors suspect is pneumonia, the latest **sickness** to beset the often **ailing** 68-year-old Russian president.

Yeltsin has been hospitalized several times in the past three years, usually with respiratory infections, including twice for pneumonia in 1997 and 1998. The Kremlin tends to hospitalize the **ailing** president at the first sign of **illness**.

Figure 3: A pair of paragraphs that contain a composite match; a word match and a WordNet match (highlighted in bold) occur within a window of five words, excluding stopwords.

paragraphs contained "common information". This was defined to be the case if the paragraphs referred to the same object and the object either (a) performed the same action in both paragraphs, or (b) was described in the same way in both paragraphs. The reviewers were then instructed to resolve each instance about which they had disagreed. It is interesting to note here that in this and subsequent annotation experiments we found significant disagreements between the judges, and large variability in their rate of agreement (kappa statistics (Carletta, 1996) between 0.08 and 0.82). The disagreement was however significantly lower when the instructions were as specific as the version above, and in any case annotators were able to resolve their differences and come with a single label of similar or not similar when they conferred after producing their individual judgments. As the above discussion illustrates, the level of similarity that we represent in our training data and that SIMFINDER tries to recover automatically is much more fine-grained than in a typical information retrieval application; we are moving from topical similarity down to the level of propositional content similarity.

We subsequently trained a classifier over both primitive and composite features using RIPPER (Cohen, 1996). RIPPER produces a set of ordered rules that can be used to judge any pair of paragraphs as similar or non-similar. Using three-fold cross-validation over the training data, RIPPER included 11 of the 43 features in its final set of rules and achieved 44.1% precision at 44.4% recall. The eleven features were Word Overlap, Proper Noun Overlap, LinkIT (noun phrase head) Overlap (Wacholder, 1998), Verb Overlap, Noun Overlap, Ad-

jective Overlap, WordNet Overlap, WordNet Verb Overlap, Verb Overlap, WordNet Collocation, and Stem Overlap (see (Hatzivassiloglou et al., 1999) for more details on the various features). The selection of eleven features rather than just words validates our claim that more than word matching is needed for effective paragraph matching for summarization. This was also verified experimentally; the standard TF*IDF measure (Salton and Buckley, 1988), which bases similarity on shared words weighted according to their frequency in each text unit and their rarity across text units, yielded 32.6% precision at 39.1% recall. We also measured the performance of a standard IR system on this task; the SMART system (Buckley, 1985), which uses a modified TF*IDF approach, achieved 34.1% precision at 36.7% recall. In all cases, we report evaluation results at the point of the precision-recall curve where precision and recall are closest, which is a summary metric combining information on the two possible kinds of errors (as 11-point precision and F-measure also do). We did not have direct access to the more recent information retrieval systems offering improvements over SMART (e.g., the TDT2 and TDT3 systems) so that we could apply them to paragraph-length text segments and directly compare their performance to our method. However, such systems still primarily use word matches for determining similarity, rely most commonly on variants of TF*IDF, and are designed to operate on text pieces much larger than sentences or paragraphs.

It is worth noting that 21 of the 43 original features were normalized according to the matching primitives' IDF scores (the number of documents in our collection they appear in). RIPPER selected none of those features, which suggests that TF*IDF is not an appropriate metric to use in evaluating similarity between small text units in a system such as ours. This observation makes sense given that in SIMFINDER the collection of documents from which document frequency is calculated has been filtered by topic and date. Thus, a primitive that would be rare in a large corpus could have an abnormally high frequency in the relatively small set of related documents on which SIMFINDER operates.

Since performing this evaluation, we have refined some of our features and added new ones. We now take a more sophisticated view of proper names, maintaining a list of previously seen proper name

	Precision	Recall	F_1-measure
Standard TF*IDF	32.6%	39.1%	35.6%
SMART	34.1%	36.7%	35.4%
1999 SIMFINDER (with RIPPER)	44.1%	44.4%	44.2%
2001 SIMFINDER (with log-linear model)	49.3%	52.9%	51.0%

Table 1: Evaluation scores for several similarity computation techniques. The test data consisted of pairs of paragraphs from closely related documents in the Reuters part of the 1997 TDT pilot corpus, manually labeled as similar or not similar.

forms and allowing for partial matches (i.e., implementing a limited co-reference resolution component) so that multiple forms of the same name can be collated. We have added filters eliminating some categories of linking verbs and function words from our feature counts, and incorporated a new feature that tracks whether two paragraphs come from the same article (hypothesizing that highly similar paragraphs are less likely to occur in the same article). Finally, we have changed our machine learning approach to allow for values of similarity in the full range between 0 and 1 rather than the "yes"/"no" decisions that RIPPER supports. Such real-valued similarities enable the clustering component of SIMFINDER to give higher weight to paragraph pairs that are more similar than others.

We use a *log-linear regression model* to convert the evidence from the various features to a single similarity value. This is similar to a standard regression model (i.e., a weighted sum of the features) but properly accounts for the changes in the output variance as we go from the normal to the binomial distribution for a response between 0 and 1 (Mc-Cullagh and Nelder, 1989). A weighted sum of the input features is used as an intermediate predictor, η, which is related to the final response R via the logistic transformation,

$$R = \frac{e^\eta}{1 + e^\eta}$$

Via an iterative process, stepwise refinement, the log-linear model automatically selects the input features that increase significantly the predictive capability of the model, thus avoiding overlearning. The model selected 7 input features, and resulted in a remarkable increase in performance over the RIPPER output (which itself offered significant improvement over standard IR methods), to 49.3% precision at 52.9% recall. Table 1 summarizes the evaluation scores obtained by the different methods,
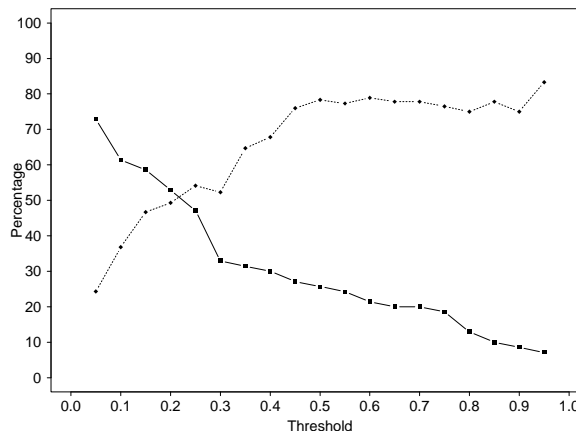

Figure 4: Precision and recall curves for the log-linear version of SIMFINDER at various decision thresholds.

while Figure 4 shows the precision-recall curves corresponding to the log-linear version at different cutoff thresholds for considering two paragraphs as similar. As in the case of the RIPPER model, the automatic selection of multiple features in the log-linear model validates our hypothesis that more than straightforward word matching is needed for effectively detecting similarity between small pieces of text.

3 Clustering Algorithm

Once similarities between any two text units have been calculated, we feed them to a clustering algorithm that partitions the text units into clusters of closely related ones. This module was added to SIMFINDER after our earlier publication of our approach to similarities, replacing an earlier heuristic placeholder, and is described in this paper for the first time. Once again we depart from traditional IR algorithms, opting instead to use an algorithm

more appropriate to the summarization task's requirements. In Information Retrieval, hierarchical algorithms such as single-link, complete-link, and groupwise-average, as well as online variants such as single pass are often used (Frakes and Baeza-Yates, 1992). Compared to non-hierarchical techniques, such algorithms trade off some of the quality of the produced clustering for speed (Kaufman and Rousseeuw, 1990), or are sometimes imposed because of additional requirements of the task (e.g., when documents must be processed sequentially as they arrive). For summarization, however, the distinctions between paragraphs are often fine-grained, and there are usually much fewer related paragraphs to cluster than documents in an IR application.

We have therefore adopted a non-hierarchical clustering technique, the *exchange method* (Späth, 1985), which casts the clustering problem as an optimization task and seeks to minimize an objective function Φ measuring the within-cluster dissimilarity in a partition $\mathcal{P} = \{C_1, C_2, \ldots, C_k\}$,

$$\Phi(\mathcal{P}) = \sum_{i=1}^{k} \left(\frac{1}{|C_i|} \sum_{\substack{x,y \in C_i \\ x \neq y}} d(x, y) \right)$$

where the dissimilarity $d(x, y)$ is one minus the similarity between x and y.

The algorithm proceeds by creating an initial partition of the text units that are to be clustered, and then looking for locally optimal moves and swaps of text units between clusters that improve Φ, until convergence is achieved. Since this is a hill-climbing method, the algorithm is called multiple times from randomly selected starting points, and the best overall configuration is selected as the final result.

We have further modified the clustering method to address some of the characteristics of data sets in summarization applications. To reduce the number of paragraphs considered for clustering, we impose an adjustable threshold on the similarity values, ignoring paragraph pairs for which their evidence of similarity is too weak. By adjusting this threshold, we can have the system create small, high-quality clusters or large, noisy clusters as needed. Since every paragraph in that filtered set is similar to at least another one, we impose an additional constraint on the clustering algorithm to never produce singleton clusters.

We also have adopted an appropriate heuristic for estimating the number of clusters for a given set of paragraphs. Since each cluster is subsequently transformed into a single sentence of the final summary, many small clusters would result in an overly lengthy summary while a few large clusters would result in a summary that omits important information. We use information on the number of links passing the similarity threshold between the clustered paragraphs, interpolating the number of clusters between the number of connected components in the corresponding graph (few clusters, for very dense graphs) and half of the number of paragraphs (lots of clusters, for very sparse graphs). In other words, the number of clusters c for a set of n text units in m connected components is determined as

$$c = m + \left(\frac{n}{2} - m \right) \left(1 - \frac{\log(L)}{\log(P)} \right)$$

where L is the observed number of links and P ($= n(n-1)/2$) is the maximum possible number of links. We use a non-linear interpolating function to account for the fact that, usually, $L \ll P$.

Partial output from a sample clustering of news paragraphs is shown in Figure 5.

4 From Clusters to Summaries

Clustering, as implemented in SIMFINDER, provides a flexible means for organizing related information in a form that can be subsequently turned into summaries of varying formats and complexities. Each cluster captures information salient to a particular facet of the input data, often a specific event, fact, or opinion. As an initial step, key terms in each cluster can be collected and used as an indicative, free-form summary (Witten et al., 1999). Alternatively, one sentence or paragraph per cluster can be selected, producing an extracted summary. These sentences can be chosen using simple positional features (e.g., the sentence located earliest in its source article) or as the centroids of their cluster (Radev et al., 2000).

We report on two specific schemas for converting the clusters to summary sentences that we have implemented at Columbia University, which nevertheless do not exhaust the possibilities. Our focus in this paper is primarily in establishing the usefulness of SIMFINDER as a component of summarization systems using variable content selection or generation back ends; hence, we do not discuss the back end systems' operation in depth.

Cluster 1	Cluster 2
MEXICO CITY (Reuters) - The United Nations' human rights chief on Wednesday said Mexico was taking steps to improve its rights problems but was still failing to bring all those responsible for abuses to justice.	"I was impressed that he (Zedillo) was not denying there were difficulties," she said.
MEXICO CITY (AP) – The top U.N. human rights official said Wednesday that attacks on activists and faulty law enforcement were among Mexico's most serious human rights woes, but she applauded its president for recognizing his country has such problems.	"He (Zedillo) was very open about there being difficulties ... I was impressed that he was not denying those difficulties," Robinson told reporters at a ceremony in which she and Mexican officials signed a letter of understanding on rights promotion.

Figure 5: Automatically produced clusters of paragraphs (partial clusters are shown).

The first of these summarization systems, CEN-TRIFUSER, utilizes SIMFINDER's output in the medical domain. In the context of the multidisciplinary Digital Library project at our site, we are looking for ways to summarize multiple medical articles for either patients or doctors. For patient-oriented summaries, CENTRIFUSER retrieves information from a number of online health resources to increase coverage, but needs SIMFINDER to unify the information and eliminate redundancy.

We take advantage of broad domain knowledge principles for the organization of the summaries, presenting information on topics such as diseases, diagnosis, and treatment separately. CENTRIFUSER stratifies the input data according to each such broad topical class; calls SIMFINDER to organize the sentences within each topic into clusters; and then picks one representative sentence from each cluster to form the final summary. Two heuristics are used for the sentence selection phase: clusters spread over multiple documents are selected first, to ensure that in a summary of limited length the most general information is included; and sentences near the start of their documents are preferred, to minimize dangling references. Figure 6 shows a summary about the heart condition "angina" produced by CENTRI-FUSER out of five related documents, each between 2,700 and 7,000 words long.

The second approach for summary generation, MULTIGEN (Barzilay et al., 1999), goes beyond sentence extraction into reformulation. Summarization by extraction has a number of well-known undesired effects (McKeown et al., 1999): sentences taken out of context often include embedded phrases that are not salient enough for a summary, may bias the summary towards a particular detail,

Treatment is designed to prevent or reduce ischemia and minimize symptoms. Angina attacks usually last for only a few minutes, and most can be relieved by rest. Most often the discomfort occurs after strenuous physical activity or an emotional upset. A doctor diagnoses angina largely by a person's description of the symptoms. The underlying cause of angina requires careful medical treatment to prevent a heart attack. Not everyone with ischemia experiences angina. If you experience angina, try to stop the activity that precipitated the attack.

Figure 6: CENTRIFUSER output for "angina."

and may create dangling references and disfluencies. For example, picking any one sentence from the cluster in Figure 7 results in the inclusion of some unnecessary details. MULTIGEN analyzes the sentences in each cluster produced by SIMFINDER and regenerates instead a new sentence containing just the information common to almost all sentences in a cluster. It operates in three phases: parsing the sentences in each cluster with an existing statistical parser (Collins, 1996), matching the central elements in the resulting dependency trees (allowing for paraphrases), and finally generating a new sentence from these matched elements. Regeneration can be achieved in two ways: Either by mapping the predicate-argument structure produced by our matching algorithm to the functional representation expected by FUF/SURGE (Elhadad, 1993; Robin, 1994) using additional constraints on real-

The quake had a magnitude of 6.9, following an earthquake in the same region in February which killed 2,300 people and left thousands homeless.

The quake registered 6.9 on the Richter scale, centered in a remote part of the country.

Contacted at his headquarters in northern Afghanistan, Abdullah said he feared thousands of people may have died in the devastating quake in northeastern Afghanistan, with a preliminary magnitude of 6.9.

(a)

The quake had a magnitude of 6.9.

(b)

Figure 7: (a) A SIMFINDER-produced cluster of similar sentences where any one of them includes unnecessary details; (b) MULTIGEN output for this cluster.

ization choice based on surface features in place of the semantic or pragmatic ones typically used in sentence generation; or by selecting a sentence from the cluster as a skeleton, and modifying it to include only phrases matched across the cluster entire while preserving the grammatical validity of the sentence. While the first approach is more general and allows us to produce more complex sentences, the second approach is robust in a noisy environment. For the example of Figure 7, both techniques would produce "The quake had a magnitude of 6.9"; note that this is explicit in the first sentence, but expressed via paraphrases such as *had* $\sim$ *registered* in the other two.

5 Conclusion

We have presented developments in our similarity module and a recently added clustering algorithm, which jointly form a flexible tool for converting textual data into groups of related text units that can be further reduced to single sentences in a summarization system. We have demonstrated quantitative improvements in performance when compared to earlier work and standard IR techniques, and shown how the information that our system produces can be used by some very different approaches to the final summary production.

We are currently focusing on extending SIM-

FINDER to multilingual features and increasing the robustness of its feature extraction process. One way to adapt our similarity model for documents in multiple languages is to re-examine the features used and select those that can be extracted from languages with less developed NLP tools than English. Resilience during translation is also a factor (we are looking at proper names, for example, as one feature that we expect would be easy to translate reliably even from minority languages). At the same time, we are testing SIMFINDER's portability in yet another domain in cooperation with the University of Massachusetts at Amherst (who provide their TDT system for initial document clustering) and MITRE, compiling an additional 15,000 of judgments on paragraph and sentence similarities for training and evaluation. We are in the process of formally measuring the effectiveness of the clustering component of SIMFINDER (Section 3) relative to the more commonly used hierarchical clustering techniques. We are also looking at ways to increase SIMFINDER's accuracy in discovering similarities that might be obscured by additional information in one or more of the matching sentences or paragraphs; for example, by clustering at the clause as well as the sentence and paragraph levels, and by reducing the relative weight of features in subordinate clauses.

References

James Allan, Jaime Carbonell, George Doddington, Jon Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Understanding and Transcription Workshop*, pages 194–218, April.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in

the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the ACL,* pages 550–557, College Park, Maryland, June.

Christopher Buckley. 1985. Implementation of the SMART information retreival system. Technical Report 85-686, Cornell University, Ithaca, New York.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98)*, pages 335–336, Melbourne, Australia, August.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, **22**(2):249–254, June.

William Cohen. 1996. Learning trees and rules with set-valued features. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96)*.

Michael Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, California.

Michael Elhadad. 1993. *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation*. Ph.D. thesis, Department of Computer Science, Columbia University, New York.

John Fiscus, George Doddington, J. Garofolo, and A. Martin. 1999. NIST's 1998 Topic Detection and Tracking evaluation (TDT2). In *Proceedings of the 1999 DARPA Broadcast News Workshop*, pages 19–24, Herndon, Virginia, February–March.

William B. Frakes and Ricardo Baeza-Yates. 1992. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Upper Saddle River, New Jersey.

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212, College Park, Maryland, June.

Vasileios Hatzivassiloglou, Luis Gravano, and Ank-

ineedu Maganti. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-00)*, pages 224–231, Athens, Greece, July.

Leonard Kaufman and Peter J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

Julian M. Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-95)*, pages 68–73, Seattle, Washington, July.

Chin-Yew Lin and Eduard Hovy. 1997. Identifying topics by position. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, pages 283–290, Washington, D.C., April.

Hans P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal*, pages 159–165.

Inderjeet Mani and Eric Bloedorn. 1997. Multi-document summarization by graph search and matching. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 622–628, Providence, Rhode Island.

Daniel Marcu. 1997. From discourse structures to text summaries. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain, August.

Peter McCullagh and John A. Nelder. 1989. *Generalized Linear Models*. Chapman and Hall, London, 2nd edition.

Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-99)*, pages 453–460, Orlando, Florida, July.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, **3**(4):235–312.

Dragomir R. Radev, Hongyan Jing, and Malgorzata

Budzikowska. 2000. Summarization of multiple documents: Clustering, sentence extraction, and evaluation. In *ANLP/NAACL Workshop on Summarization*, Seattle, Washington, April.

Jacques Robin. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. Ph.D. thesis, Department of Computer Science, Columbia University, New York.

Gerard Salton and Christopher Buckley. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, **25**(5):513–523.

Gerard Salton, Amit Singhal, Mandar Mitra, and Christopher Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, **33**(2):193–207.

Helmuth Späth. 1985. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples*. Ellis Horwood, Chichester, West Sussex, England.

Nina Wacholder. 1998. Simplex NPs clustered by head: A method for identifying significant topics in a document. In *Proceedings of the Workshop on the Computational Treatment of Nominals*, pages 70–79, Montreal, Canada, October. COLING-ACL.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *Proceedings of the ACM Conference on Digital Libraries (DL-99)*.