Latent Semantic Analysis and the Construction of Coherent Extracts

Tristan Miller
German Research Center for Artificial Intelligence[*]
Erwin-Schrödinger-Straße 57, 67663 Kaiserslautern
tristan.miller@dfki.de

Keywords: automatic summarization, latent semantic analysis, LSA, coherence, extracts

Abstract

We describe a language-neutral automatic summarization system which aims to produce coherent extracts. It builds an initial extract composed solely of topic sentences, and then recursively fills in the topical lacunae by providing linking material between semantically dissimilar sentences. While experiments with human judges did not prove a statistically significant increase in textual coherence with the use of a latent semantic analysis module, we found a strong positive correlation between coherence and overall summary quality.

1 Introduction

A major problem with automatically-produced summaries in general, and extracts in particular, is that the output text often lacks fluency and organization. Sentences often leap incoherently from topic to topic, confusing the reader and hampering his ability to identify information of interest. Interest in producing textually coherent summaries has consequently increased in recent years, leading to a wide variety of approaches, including IR-influenced techniques (Salton *et al.* [1997]; Carbonell and Goldstein [1998]), variations on lexical chaining (Brunn *et al.* [2002]; Karamuftuoglu [2002]), and discourse structure analysis (Marcu [1997, 1999]; Chan *et al.* [2000]). Unfortunately, many of these techniques are tied to a particular language or require resources such as a list of discourse keywords and a manually marked-up corpus; others are constrained in the type of summary they can generate (*e.g.*, general-purpose *vs.* query-focussed).

In this paper, we present a new, recursive method for automatic text summarization which aims to preserve both the topic coverage and the coherence of the source document, yet has minimal reliance on language-specific NLP tools. Only word- and sentence-boundary detection routines are required. The system produces general-purpose extracts of single documents, though it should not be difficult to adapt the technique to query-focussed summarization, and may also be of use in improving the coherence of multi-document summaries.

2 Latent semantic analysis

Our system fits within the general category of IR-based systems, but rather than comparing text with the standard vector-space model, we employ latent semantic analysis (LSA) [Deerwester *et al.*, 1990], a technique originally developed to circumvent the problems of synonymy and polysemy in IR. LSA extends the traditional vector-space document model with *singular value decomposition*, a process by which the term–sentence co-occurrence matrix representing the source document is factored into three smaller matrices of a particular form. One such matrix is a diagonal matrix of *singular values*; when one or more of the smallest singular values are deleted and the three matrices multiplied together, the product is a least-squares best fit to the original matrix. The apparent result of this smearing of values is that the ap-

[*] The research described in this paper was carried out while the author was at the University of Toronto.

proximated matrix has captured the latent transitivity relations among terms, allowing for identification of semantically similar sentences which share few or no common terms withal. We believe that the deep semantic relations discovered by LSA may assist in the identification and correction of abrupt topic shifts between sentences.

3 Algorithm

The input to our summarizer is a plain text document, which is converted into a list of tokenized sentences. A tokenizer and sentence-boundary disambiguation algorithm may be used for these first steps.

The list of m sentences (indexed from 1 to m) is then segmented into linearly discrete topics. This can be done manually if the original document is structured (*e.g.*, a book with chapters, or an article with sections), or a linear text segmentation algorithm, such as C99 [Choi, 2000] can be used. The output of this step is a list of sentence indices $\langle t_1, \ldots, t_{n+1}\rangle$, where, for the ith of the n topics, t_i is the index of the first sentence of the topic segment and $t_{i+1} - 1$ is the index of the last sentence of the topic segment. We stipulate that there are no sentences which do not belong to a topic segment, so for all t_i, we have $t_i < t_{i+1}$, and

$$t_i = \begin{cases} 1 & \text{if } i = 1; \\ m+1 & \text{if } i = n+1; \\ \begin{array}{l}\text{index of first sentence}\\ \text{of the } i\text{th topic}\end{array} & \text{otherwise.} \end{cases}$$

As mentioned previously, we use LSA to measure semantic similarity, so before we can begin constructing the extract, we need to construct a reduced-dimensionality term–sentence co-occurrence matrix. Once this is done, a preliminary extract is produced by choosing a representative "topic sentence" from each segment—that is, that sentence which has the highest semantic similarity to all other sentences in its topic segment. These topic sentences correspond to a list of sentence indices $\langle r_1, \ldots, r_n\rangle$ such that

$$r_i = \operatorname*{arg\,max}_{t_i \le j < t_{i+1}} \sum_{k=t_i}^{t_{i+1}-1} \mathrm{sim}\,(j,k)\,,$$

where $\mathrm{sim}\,(x,y) \in [-1,1]$ is the LSA cosine similarity score for the sentences with indices x and y. In order to preserve important information which may be found at the beginning of the document, and also to account for the possibility that the document contains only one topic segment, we always consider the first sentence of the document to be a topic sentence—*i.e.*, $r_0 = 1$—and include it in our initial extract.[1] Let us refer to this initial extract as $E_0 = \langle e_{0,1}, \ldots, e_{0,n+1}\rangle$ where $e_{0,i} = r_{i-1}$.

As we might imagine, this basic extract will have very poor coherence, since every sentence addresses a completely different topic. However, we can improve its coherence by selecting from the set $\langle 1, \ldots, m\rangle \setminus E_0$ a number of indices for "glue" sentences between adjacent pairs of sentences represented in E_0. We consider an appropriate glue sentence between two others to be one which occurs between them in the source document, and which is semantically similar to both. Thus we look for sentence indices $G_1 = \langle g_{1,1}, \ldots, g_{1,n}\rangle$ such that

$$g_{1,i} = \operatorname*{arg\,max}_{e_{0,i} < j < e_{0,i+1}} f\left(\mathrm{sim}'\,(j, e_{0,i})\,, \mathrm{sim}'\,(j, e_{0,i+1})\right),$$

where

$$f\,(x,y) = xy \cdot (1 - |x-y|)$$

and

$$\mathrm{sim}'\,(x,y) = \begin{cases} 0 & \text{if } \mathrm{sim}\,(x,y) > \alpha; \\ 0 & \text{if } \mathrm{sim}\,(x,y) < 0; \\ \mathrm{sim}\,(x,y) & \text{otherwise.} \end{cases}$$

for $\alpha \in [0,1]$. The purpose of $f()$ is to reward glue sentences which are similar to their boundary sentences, but to penalize if the similarity is too biased in favour of only one of the boundaries. The revised similarity measure $\mathrm{sim}'()$ ensures that we do not select a glue sentence which is nearly equivalent to any one boundary—such a sentence is redundant. (Of course, useful values of α will be 1 or close thereto.)

Once we have G_1, we can construct a revised extract $E_1 = \langle e_{1,1}, \ldots, e_{1,2n+1}\rangle = \langle E_0 \cup G_1\rangle$.[2]

[1] In practice, it may be the case that $r_1 = 1$, in which case inclusion of r_0 is not necessary. In this paper we assume, without loss of generality, that $r_1 \neq 1$.

[2] For notational convenience, we take it as understood that the sentence indices in the extracts E_i are sorted in ascending order—that is, $e_{i,j} < e_{i,j+1}$ for $1 \le j < |E_i|$.

More generally, however, we can repeat the gluing process recursively, using E_i to generate G_{i+1}, and hence E_{i+1}. The question that arises, then, is when to stop. Clearly there will come a point at which some $e_{i,j} = e_{i,j+1} - 1$, thus precluding the possibility of finding any further glue sentences between them. We may also encounter the case where for all k between $e_{i,j}$ and $e_{i,j+1}$, $f\left(\mathrm{sim}'\left(k, e_{i,j}\right), \mathrm{sim}'\left(k, e_{i,j+1}\right)\right)$ is so low that the extract's coherence would not be significantly improved by the addition of an intermediary sentence. Or, we may find that the sentences with indices $e_{i,j}$ and $e_{i,j+1}$ are themselves so similar that no glue is necessary. Finally, it is possible that the user wishes to constrain the size of the extract to a certain number of sentences, or to a fixed percentage of the original document's length. The first of these stopping conditions is straightforward to account for; the next two can be easily handled by introducing two fixed thresholds β and γ: when the similarity between adjacent sentences from E_i exceeds β, or when the value of $f()$ falls below γ, no glue sentence is suggested for the pair in question.

The case of maximum summary length is a bit trickier. If we are not concerned about undershooting the target length ℓ, then we can simply halt the algorithm once $|E_i| \geq \ell$, and then take E_{i-1} (or E_i, if $|E_i| = \ell$) as the final extract. Most real-world applications, however, demand that we maximize the extract size. Given E_{i-1} of length $\ell - p$, the optimal extract E of length ℓ is the one which glues together the p largest gaps in E_{i-1}.

A version of the gluing algorithm which takes into account all four stopping conditions is shown in Algorithm 1.

Once the final set of sentences for the extract has been selected, we send the sentences, in their original order of occurrence, to the topic segmenter. The discovered topic segments are then used by a simple text formatter to partition the summary into sections or paragraphs for easy reading.

3.1 Complexity analysis

Given an initial extract of length n, the first recursion of Algorithm 1 will add at most $n - 1$ sentences to the extract, yielding a new extract of length $2n - 1$. In general, at most $2^{i-1}n$ sentences will be added on the ith recursion, bringing the extract length to $2^i n - 1$ sentences. Therefore, to achieve an extract of length $\ell > n$, the algorithm needs to recurse at least

$$\left\lceil \log_2 \frac{\ell + 1}{n} \right\rceil$$

times. The worst case occurs when $n = 2$ and the algorithm always selects a glue sentence which is adjacent to one of the boundary sentences (with indices e_1 and e_2). In this case, the algorithm must recurse $\min\left(\ell, e_2 - e_1\right)$ times, which is limited by the source document length, m.

On each recursion i of the algorithm, the main loop considers at most $m - \left(2^i n - 1\right)$ candidate glue sentences, comparing each one with two of the $2^i n - 1$ sentences already in the extract. To simplify matters, we note that $2^i n - 1$ can never exceed m, so the number of comparisons must be, at worst, proportional to m. The comparison function, $\mathrm{sim}()$, runs in time proportional to the number of word types, w, in the original document. Thus an upper bound on the time complexity of a naïve implementation of Algorithm 1 is $O(wm^2)$.

Running time can be cut down considerably in the general case, however. Since $\mathrm{sim}(i, j)$ remains constant, we can save time by precomputing a triangular similarity matrix of all pairs of sentences in the document, or better yet, by using memoization (*i.e.*, caching intersentential similarity values as they are computed). The algorithm could be further improved by having the loop skip over adjacent extract sentences for which no glue was found on a previous recursion. At any rate, the running time of the summarizer as a whole will likely be dominated by the singular value decomposition step of the LSA stage (at least $O(wm^2)$) and possibly too by the topic segmenter (for C99, also $O(wm^2)$).

4 Evaluation

In general there are two approaches to evaluating summaries: *intrinsic* evaluations, which rate the summary in and of itself, and *extrinsic* evaluations, which test the summary in relation to some other task [Spärck Jones and Galliers, 1996]. Popular intrinsic approaches include *quality evalua-*

Algorithm 1: glue()

```
input        : initial extract $E$, maximum extract length $\ell$
output       : largest coherent extract of length $\leq \ell$
precondition : $|E| < \ell$
assumption   : Lists are kept sorted in ascending order. Where list elements are coordinate pairs,
               the sorting key is the first coordinate.
$G \leftarrow \langle\rangle$;
for $i \leftarrow 1$ to $|E| - 1$ do
    $s \leftarrow \mathrm{sim}(E[i], E[i+1])$;
    if $E[i] = E[i+1] - 1$ or $s > \beta$ then continue;
    $g \leftarrow \underset{E[i]<j<E[i+1]}{\arg\max} \; f(\mathrm{sim}'(j, E[i]), \mathrm{sim}'(j, E[i+1]))$;
    if $f(\mathrm{sim}'(g, E[i]), \mathrm{sim}'(g, E[i+1])) \geq \gamma$ then $G \leftarrow G \cup \langle (s, g) \rangle$;
end
if $|G| = 0$ then
    return $E$;
else if $|E| + |G| \geq \ell$ then
    return $E \cup \left\langle x \;\middle|\; (y, x) \in \bigcup_{i=|E|+|G|-\ell+1}^{|G|} G[i] \right\rangle$;
else
    return $\mathrm{glue}(E \cup \langle x \mid (y, x) \in G \rangle, \ell)$;
end
```

tion, where human graders grade the summary in isolation on the basis of relevance, grammaticality, readability, *etc.*; and *gold-standard comparison*, where the summary is compared (by humans or automatically) with an "ideal" summary. Extrinsic methods are usually domain- or query-dependent, but two popular methods which are relatively generic are *relevance assessment*, where the summarizer acts as the back-end to an information retrieval system, and *reading comprehension*, where the summaries are used as input to a question-answering task. In both cases the idea is to compare performance of the task given the summaries versus the whole documents.

Though it could be argued that reading comprehension is somewhat dependent on coherence, almost all evaluation methods are designed primarily to assess topic coverage and information relevance. This may be because to date, researchers have concentrated on evaluation of highly-compressed summaries, where coherence necessarily takes a back seat to topic coverage. Another reason why coherence is not measured directly is the dearth of good, automatable evaluation metrics for the trait. One approach commonly used in essay assessment [Miller, 2003a] is to average the semantic similarity (using the cosine coefficient, with or without LSA) of all adjacent sentence pairs. This technique is not appropriate for our algorithm because by definition its summaries are guaranteed to have good intersentential cosine scores. This approach has the additional disadvantage of rewarding redundancy.

A more recent approach to automated coherence assessment is to check for the presence or absence of discourse relations [Marcu, 2000]. The problem with this approach is that the vast majority of discourse relations are not signalled by an obvious discourse marker [Marcu and Echihabi, 2002].

Since we also could not come up with a new task-based evaluation which would measure coherence in isolation, we felt we were left with no choice but to use the intrinsic method of quality evaluation. We therefore recruited human judges to provide ratings for our summaries' coherence, and for the sake of convenience and simplicity, we also used them to assess other aspects of summary quality.

4.1 Experiment

Source data We had hoped to use the TIPSTER documents commonly used in summary evaluations at the annual Document Understanding Conference (DUC). However, most of them were very short and focussed on single, narrow topics, making them unsuitable for an evaluation of summary coherence. We therefore randomly selected one 1000-word and one 2000-word article from a current encyclopedia, plus one of the five longest newspaper articles from the DUC 2001 trial data.

Comparison systems On the basis of our own informal observations, we determined that our system (hereinafter `lsa`) performed best with a retention of 20–30% of the singular values and thresholds of $\alpha = 0.9$, $\beta = 1.0$, and $\gamma = 0.1$. More parsimonious cutoffs tended to result in summaries greatly in deficit of the allowed length.

We selected four third-party comparison systems based on their availability and similarity to our own technique and/or goals: Microsoft Word, commonly available and therefore an oft-used benchmark; Lal and Rüger [2002], a Bayesian classifier summarizer intended to assist students with reading comprehension; Copernic, a commercial summarizer based partly on the work of Turney [2000]; and Sinope (formerly Sumatra), which, like `lsa`, employs a technique for identifying latent semantic relations [Lie, 1998]. In our results tables we refer to these systems as `word`, `plal`, `copernic`, and `sinope`, respectively.

Baselines There are two popular methods for constructing baseline extracts of a given length, both of which are used in our study. The first (`random`) is to randomly select n sentences from the document and present them in their original order of appearance. The second way (`init`), based on the observation that important sentences are usually located at the beginning of paragraphs, is to select the initial sentence of the first n paragraphs.

In order to measure the contribution of LSA to our system's performance, we also employed a version of our summarizer (`nolsa`) which does not use the singular value decomposition module.

Test procedure We ran the eight summarizers on the three source documents twice each—once to produce a "short" summary (around 100 words) and once to produce a "long" summary (around 300 words). We then recruited human judges who self-identified as fluent in English, the language of the source documents. The judges were provided with these documents and the 48 summaries grouped according to source document and summary length. Within each document–summary length group, the summaries were labelled only with a random number and were presented in random order. We asked the judges to read each source document and then assign to each of its summaries an integer score ranging from 1 (very poor) to 5 (very good) on each of three dimensions: comprehensiveness (*i.e.*, topic coverage), coherence, and overall quality. The judges were given the compression ratio for each summary and told to take it under consideration when assigning their ratings.

4.2 Results

4.2.1 Interjudge agreement

To compare interjudge agreement, we computed correlation matrices for each of coherence, comprehensiveness, and overall quality ratings. Interjudge agreement on coherence was generally low, with the mean Pearson correlation coefficient r ranging from 0.0672 to 0.3719. Agreement on comprehensiveness and quality was better, but still only moderate, with r in the ranges $[0.2545, 0.4660]$ and $[0.2250, 0.4726]$, respectively. Why the correlation is only moderate is difficult to explain, though given the similarly low agreement in the DUC 2001 evaluations [Lin and Hovy, 2002], it was not entirely unexpected. Though we had made an effort to narrowly define coherence in the written instructions to the judges, it is possible that some of them nevertheless conflated the term with its more conventional meaning of intelligibility, or with cohesion. As discussed in Miller [2003b], this last possibility seems to be supported by the judges' written comments.

4.2.2 Comparative performance of summarizers

We used SAS to perform a three-way repeated-measures analysis of variance (ANOVA) for each of the three dimensions: coherence, comprehensiveness, and overall quality. Quite unexpectedly, the (*document*, *summary length*, *summarizer*) three-way interaction effect was significant at the 0.05 confidence level for all three dimensions ($p = 0.0151$, $p < 0.0001$, and $p = 0.0002$, respectively). This means it would have been very difficult, if not impossible, to make any generalizations about the performance of the individual summarizers. On the assumption that the type of document was irrelevant to summarizer performance, we added the document scores for each (*summarizer*, *summary length*, *rater*) triplet to get new coherence, comprehensiveness, and overall quality measurements in the range $[3, 15]$. We then performed two-way repeated-measures ANOVAs for each dimension. The two-way interaction effect was still significant for comprehensiveness ($p = 0.0025$) and overall quality ($p = 0.0347$), but not for coherence ($p = 0.6886$).

Coherence In our coherence ANOVA, the only significant effect was the summarizer ($p < 0.0001$). That summary length was not found to be significant ($p = 0.0806$) is somewhat surprising, since we expected a strong positive correlation between the coherence score and the compression ratio. Though we did ask our judges to account for the summary length when assigning their scores, we did not think that very short extracts could maintain the same level of coherence as their longer counterparts. It may be that summary length's effect on coherence is significant only for summaries with much higher compression ratios than those used in our study.

With respect to the comparative performance of the summaries, only 7 of the 28 pairwise comparisons from our ANOVA were significant at the 0.05 confidence level. The initial-sentences baseline was found to perform significantly better than every other summarizer ($p \leq 0.0008$[3]) except `copernic` and `plal`. The only other significant result we obtained for coherence was that the `sinope` summarizer performed worse than `copernic` ($p = 0.0050$) and `plal` ($p = 0.0005$). Using these pairwise comparisons, we can partition the summarizers into three overlapping ranks as shown in Table 1.

Rank(s)		Summarizer	Mean rating
A		`init`	11.1111
A	B	`plal`	9.9722
A	B	`copern`	9.6667
C	B	`word`	8.9444
C	B	`lsa`	8.7222
C	B	`nolsa`	8.6667
C	B	`random`	8.4722
C		`sinope`	7.7500

Table 1: Summarizer coherence rankings

Comprehensiveness and overall quality The mean comprehensiveness score for long summaries was higher than that for short summaries by a statistically significant 1.9792 ($p < 0.0001, \alpha = 0.05$). In fact, in no case did any summarizer produce a short summary whose mean score exceeded that of the long summary for the same document. This could be because none of the short summaries covered as many topics as our judges thought they could have, or because the judges did not or could not completely account for the compression level. In order to resolve this question, we would probably need to repeat the experiment with abstracts produced by human experts, which presumably have optimal comprehensiveness at any compression ratio.

Likewise, the overall quality scores were dependent not only on the summarizer but also on the summary length, but it is not clear whether this is because our judges did not factor in the compression ratio, or because they genuinely believed that the shorter summaries were not as useful as they could have been for their size.

As with coherence, we can partition the summarizers into overlapping ranks based on their statistically significant scores. Because the (*summary length*, *summarizer*) interaction was significant,

[3] All p values in this chapter from here on are Tukey-adjusted.

we produce separate rankings for short and long summaries. (See Tables 2 and 3.)

4.2.3 Relationship among dimensions

Intuition tells us that overall quality of a summary depends in part on both its topic flow and its topic coverage. To see if this assumption is borne out in our data, we calculated the Pearson correlation coefficient for our 864 pairs of coherence–overall quality ratings and comprehensiveness–overall quality ratings. The correlation between coherence and overall quality was strong at $r = 0.6842$, and statistically significant ($t = 27.55$) below the 0.001 confidence level. The comprehensiveness–overall quality correlation was also quite strong ($r = 0.7515, t = 33.44, \alpha < 0.001$).

4.3 Analysis

Unfortunately, moderate to low interjudge agreement for all three dimensions, coupled with an unexpected three-way interaction between the summarizers, the source documents, and the compression ratio, stymied our attempts to make high-level, clear-cut comparisons of summarizer performance. The statistically significant results we did obtain have confirmed what researchers in automatic summarization have known for years: that it is very hard to beat the initial-sentences baseline. This baseline consistently ranked in the top category for every one of the three summary dimensions we studied. While the `copern` and `plal` systems sometimes had higher mean ratings than `init`, the difference was never statistically significant.

The performance of our own systems was unremarkable; they consistently placed in the second of the two or three ranks, and only once in the first as well. Though one of the main foci of our work was to measure the contribution of the LSA metric to our summarizer's performance, we were unable to prove any significant difference between the mean scores for our summarizer and its non-LSA counterpart. The two systems consistently placed in the same rank for every dimension we measured, with mean ratings differing by no more than 6%. As a case study in Miller [2003b] suggests, this nebulous result may be due more to the LSA summarizer's unfortunate choice of topic sentences than to its gluing process, which actually seemed to perform well with the material it was given.

5 Conclusion

Our goal in this work has been to investigate how we can improve the coherence of automatically-produced extracts. We developed and implemented an algorithm which builds an initial extract composed solely of topic sentences, and then fills in the lacunae by providing linking material between semantically dissimilar sentences. In contrast with much of the previous work we reviewed, our system was designed to minimize reliance on language-specific features.

Our study revealed few clearly-defined distinctions among the summarization systems we reviewed, and no significant benefit to using LSA with our algorithm. Though our evaluation method for coherence was intended to circumvent the limitations of automated approaches, the use of human judges introduced its own set of problems, foremost of which was the low interjudge agreement on what constitutes a fluent summary. Despite this lack of consensus, we found a strong positive correlation between the judges' scores for coherence and overall summary quality. We would like to take this as good evidence that the production of coherent summaries is an important research area within automatic summarization. However, it may be that humans simply find it too difficult to evaluate coherence in isolation, and end up using other aspects of summary quality as a proxy measure.

Acknowledgments

This research was supported in part by NSERC of Canada. Thanks to Graeme Hirst and Gerald Penn for their advice.

References

Meru Brunn, Yllias Chali, and Barbara Dufour. UofL summarizer at DUC 2002. In *Workshop on Automatic Summarization, ACL 2002*, volume 2, pages 39–44, July 2002.

J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR 98*, pages 335–336, August 1998.

Short summaries

Rank(s)		Summarizer	Mean rating
A		`copern`	10.0556
A		`plal`	9.6667
A	B	`init`	8.5556
A	B	`nolsa`	8.1111
	B	`lsa`	7.5556
C	B	`sinope`	7.0000
C	B	`word`	6.9444
C		`random`	5.3889

Long summaries

Rank(s)		Summarizer	Mean rating
A		`plal`	11.9444
A	B	`copern`	10.5556
A	B	`init`	10.2222
	B	`sinope`	9.6667
	B	`word`	9.6111
	B	`random`	9.2222
	B	`lsa`	8.9444
	B	`nolsa`	8.9444

Table 2: Summarizer comprehensiveness rankings

Short summaries

Rank(s)		Summarizer	Mean rating
A		`copern`	9.7222
A	B	`init`	9.4444
A	B	`plal`	9.0556
A	B	`nolsa`	7.5000
C	B	`lsa`	7.3333
C		`word`	6.9444
C		`sinope`	6.7778
C		`random`	5.5556

Long summaries

Rank(s)		Summarizer	Mean rating
A		`plal`	11.1667
A	B	`init`	10.2778
A	B	`copern`	9.9444
A	B	`word`	9.2222
A	B	`lsa`	9.0556
	B	`random`	8.5000
	B	`nolsa`	8.3333
	B	`sinope`	8.1667

Table 3: Summarizer overall quality rankings

W. K. Chan, T. B. Y. Lai, W. J. Gao, and B. K. T'sou. Mining discourse markers for Chinese textual summarization. In *Workshop on Automatic Summarization, ACL 2000*, pages 11–20, 2000.

Freddy Choi. Advances in domain-independent linear text segmentation. In *NAACL 2000 and the 6th ACL Conference on Applied NLP*, pages 26–33, April 2000.

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

Murat Karamuftuoglu. An approach to summarisation based on lexical bonds. In *Workshop on Automatic Summarization, ACL 2002*, volume 2, pages 86–89, July 2002.

Partha Lal and Stefan Rüger. Extract-based summarization with simplification. In *Workshop on Automatic Summarization, ACL 2002*, volume 2, pages 90–96, July 2002.

D. H. Lie. Sumatra: a system for automatic summary generation. In *14th Twente Workshop on Language Technology*, December 1998.

Chin-Yu Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Workshop on Automatic Summarization, ACL 2002*, volume 1, pages 45–51, July 2002.

Daniel Marcu. From discourse structures to text summaries. In *Workshop on Intelligent Scalable Text Summarization, ACL97 and the EACL97*, pages 82–88, July 1997.

Daniel Marcu. Discourse trees are good indicators of importance in text. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 123–136. MIT Press, Cambridge, 1999.

Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, November 2000.

Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *ACL 2002*, pages 368–375, July 2002.

Tristan Miller. Essay assessment with latent semantic analysis. *Journal of Educational Computing Research*, 28(3), 2003a.

Tristan Miller. Generating coherent extracts of single documents using latent semantic analysis. Master's thesis, University of Toronto, March 2003b.

G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–207, 1997.

Karen Spärck Jones and Julia Rose Galliers. *Evaluating Natural Language Processing Systems*. Lecture Notes in Artificial Intelligence 1083. Springer, Berlin, 1996.

P. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.