Latent Semantic Analysis for Text Segmentation

Freddy Y. Y. Choi
Artificial Intelligence Group
Dept. of Computer Science
University of Manchester
Manchester M13 0AY, UK
choif@cs.man.ac.uk

Peter Wiemer-Hastings
HCRC
University of Edinburgh
2, Buccleuch Place
Edinburgh EH8 9LW, UK
peterwh@cogsci.ed.ac.uk

Johanna Moore
HCRC
University of Edinburgh
2, Buccleuch Place
Edinburgh EH8 9LW, UK
jmoore@cogsci.ed.ac.uk

Abstract

This paper describes a method for linear text segmentation that is more accurate or at least as accurate as state-of-the-art methods (Utiyama and Isahara, 2001; Choi, 2000a). Inter-sentence similarity is estimated by latent semantic analysis (LSA). Boundary locations are discovered by divisive clustering. Test results show LSA is a more accurate similarity measure than the cosine metric (van Rijsbergen, 1979).

1 Introduction

The aim of linear text segmentation is to partition a document into blocks, such that each segment is coherent and consecutive segments are about different topics. This procedure is useful in information retrieval (Hearst and Plaunt, 1993; Hearst, 1994; Yaari, 1997; Reynar, 1999), summarisation (Reynar, 1998), text understanding, anaphora resolution (Kozima, 1993), language modelling (Morris and Hirst, 1991; Beeferman et al., 1997) and text navigation (Choi, 2000b).

This paper presents a new algorithm for segmenting written text. The method builds on previous work by Choi (2000a). The primary distinction is the use of latent semantic analysis (LSA) in formulating the similarity matrix. We discovered that (1) LSA is a more accurate measure of similarity than the cosine metric, (2) stemming does not always improve segmentation accuracy and (3) ranking is crucial to cosine but not LSA.

2 Background

A text segmentation algorithm has three main parts. First, the input text is divided into elementary blocks. Second, a similarity metric identifies blocks that are about the same topic. Finally, topic boundaries are discovered by a clustering algorithm.

2.1 Elementary block

An elementary block is the smallest text segment that can describe an entire topic, e.g. sentences (Ponte and Croft, 1997), paragraphs (Yaari, 1997) and arbitrary-sized segments (Hearst, 1994). Linguistic theories (Chafe, 1979; Longacre, 1979; Kieras, 1982) and work in information retrieval (Salton et al., 1993; Kaszkiel and Zobel, 1997) suggest a coherent text segment is represented by paragraphs. We argue that a paragraph can address multiple topics and is motivated by content, writing style and presentation. Thus, a topic segment is a collection of sentences. This view is supported by previous work in text segmentation (Ponte and Croft, 1997; Choi, 2000a).

2.2 Similarity metric

A similarity metric estimates the likelihood of two segments describing the same topic. Existing methods fall into one of two categories. Lexical cohesion methods stem from the work of Halliday and Hasan (1976), in which a coherent topic segment is believed to contain parts with similar vocabulary. Implementations of this use word stem repetition (Youmans, 1991; Reynar, 1994; Ponte and Croft, 1997), context vectors (Hearst, 1994; Yaari, 1997; Kaufmann, 1999; Eichmann et al., 1999; Choi, 2000a), entity repetition (Kan et al., 1998), thesaurus relations (Morris and Hirst, 1991), spread activation over dictionary (Kozima, 1993), word distance model (Beeferman et al., 1997) and word frequency model (Reynar, 1999; Utiyama and Isahara, 2001) to detect cohesion. These methods are typically applied in information retrieval (Hearst, 1994; Reynar, 1998) to segment written text.

Multi-source methods use cue phrases, prosodic features, ellipsis, anaphora, syntactic features, language models and lexical cohesion metrics to detect topic boundaries. Features are combined using decision trees (Miike et al., 1994; Kurohashi and Nagao, 1994; Litman and Passonneau, 1995), probabilistic models (Hajime et al., 1998) and maximum entropy models (Beeferman et al., 1997; Reynar, 1998). The aim is to improve segmentation accuracy by combining multiple indicators of topic shift. These methods are typically applied in topic detection and tracking (Allan et al., 1998) to segment transcribed text and broadcast news stories.

2.3 Clustering

Topic boundaries are discovered by merging consecutive elementary blocks that are about the same topic. Existing algorithms used a sliding window (Hearst, 1994), lexical chains (Morris, 1988; Kan et al., 1998), dynamic programming (Ponte and Croft, 1997; Heinonen, 1998; Utiyama and Isahara, 2001), agglomerative clustering (Yaari, 1997) and divisive clustering (Reynar, 1994; Choi, 2000a) to determine the optimal segmentation. The main difficulty in clustering is automatic termination, i.e. determining the number of topic boundaries in a document.

3 A new method

The input to our algorithm is a list of tokenised sentences $S = \{s_1, .., s_n\}$. Content words are identified by removing punctuation marks and stopwords from S. A term frequency vector f_i is then constructed for each sentence i. f_{ij} denotes the number of times content word j occurs in s_i.

3.1 Inter-sentence similarity in C99

The C99 algorithm (Choi, 2000a) uses the cosine metric (van Rijsbergen, 1979) (eq. 1) to compute a $n \times n$ similarity matrix M for S. M_{ij} represent the similarity between s_i and s_j. The assumption is, two sentences with similar word usage are likely to be about the same topic. This idea has two main problems. First, the estimate is inaccurate for short passages. Second, synonyms are considered negative evidence, e.g. *car* $\in s_i$ and *automobile* $\in s_j$ implies s_i and s_j are dissimilar.

$$M_{ij} = cos(f_i, f_j) = \frac{\sum_k f_{ik} \times f_{jk}}{\sqrt{\sum_k f_{ik}^2 \times \sum_k f_{jk}^2}} \qquad (1)$$

The first problem was addressed by replacing M_{ij} with its *rank* R_{ij} (eq. 2, r defines the local context). The idea is, the difference in magnitude is inaccurate, thus one can only use the order as evidence for segmentation. Lets consider $X = \{x_1, x_2, x_3\} = \{1, 3, 6\}$ as the length of three objects. If X was measured with an ordinary ruler, one can conclude that x_2 is three times longer than x_1. This is a *quantitative* analysis of X, i.e. the quantity is significant. However, if the ruler was warped, but the order of the markings is preserved, one can only conclude that $x_1 < x_2 < x_3$. This is a *qualitative* analysis of X, i.e. the order is significant but the relative value has no meaning. This is a more robust interpretation of X.

$$R_{ij} = \frac{|\forall p, q \in \{-r, ..., r\} : M_{ij} > M_{pq}|}{(2r+1)^2} \qquad (2)$$

The second problem was addressed by applying a stemming algorithm (Porter, 1980) to S, such that syntactically motivated inflections are placed in an equivalent class. For example, *cooking, cooked, cooks, cooker* are all instances of the class *cook.* Unlike morphological analysers (Koskenniemi, 1983, for example), a stemming algorithm does not identify the morphemes. Its simply removes common affixes from a word, e.g. *combines, combine* → *combin, depart, department* → *depart.* Thus, similar surface forms are considered positive evidence in the similarity estimate. We propose that latent semantic analysis offers a better solution to the term matching problem.

3.2 Latent semantic analysis

LSA (Deerwester et al., 1990) stems from work in information retrieval, where the main difficulty is formulating a similarity metric that associates a user query with the relevant documents in a database. The basic keyword search approach retrieves all documents which contain some or all of the query terms. This is inaccurate since the same concept may be described using different terms. To circumvent this, Jing and Croft (1994) developed an association thesaurus for matching semantically related words.

Xu and Croft (1996) offered a train-able method call *local context analysis* (LCA) which replaces each query term with frequently co-occurring words. Roughly speaking, LCA computes a word co-occurrence matrix C for a training corpus. A threshold is then applied such that large values in C are replaced by 1 and other values become 0. Each row C_i can be considered as a feature vector for word i. The meaning of a text is approximated by the sum of the word feature vectors. Similarity between two texts is estimated by the distance between the corresponding feature vectors (Ponte and Croft, 1997, for details).

LSA is a classification approach to query expansion. The method is similar to LCA in that the "meaning" of a word w is represented by its relation to other words. The primary distinction is, LSA applies principle component analysis to a word similarity matrix to identify the best features for distinguishing dissimilar words. Like LCA, the meaning of a text is computed as the sum of the word feature vectors. Text similarity is measured by the cosine of the corresponding feature vectors. Although LSA is not necessarily the most effective similarity metric for information retrieval, it remains of interest since it has been shown to match human similarity judgements on a wide range of tasks (Landauer and Dumais, 1997; Wolfe et al., 1998; Wiemer-Hastings et al., 1999, for example).

3.2.1 Training LSA

LSA is trained on a set of texts $\Delta = \{\delta_1, ..., \delta_m\}$ with vocabulary $\{w_1, ..., w_n\}$. A $n \times m$ matrix A is calculated, in which, A_{ij} is the number of times w_i

occurs in δ_j. The values are scaled according to a general form of inverse document frequency,

$$B_{ij} = A_{ij} \times \frac{m}{|\forall k \in \{1, ..., m\} : A_{ik} > 0|}$$

Singular value decomposition, or SVD (Golub and van Loan, 1989) is then applied to yield $B = U\Sigma V^T$, where X^T denotes the transposed matrix of X. The columns of U and V are the eigenvectors of BB^T and B^TB, respectively. The diagonal values of Σ are the corresponding singular values, i.e. the non-negative square roots of the eigenvalues of BB^T. These are sorted in descending order.

$W = BB^T$ is a word similarity matrix, where W_{ij} is the dot-product of rows B_i and B_j, i.e. an estimate of the similarity between w_i and w_j. Lets consider each column in W as a feature. The "meaning" of w_i is expressed in terms of its similarity to $\{w_1, .., w_n\}$, i.e. row i of W is a feature vector for w_i. As a classification problem, the eigenvectors of W are the principle axis for distinguishing the feature vectors. In another word, a $n \times k$ matrix Λ_k which consists of the first k columns of U is the best approximation of W in $k-$dimensional space. Λ_k is referred to as the $k-$dimensional LSA space for Δ. $\Lambda_k(i)$ is the LSA feature vector for word w_i, i.e. the $i-$th row in Λ_k.

Applying SVD to W has three main benefits. First, Λ_k is a concise representation of W. Thus, storage and computational complexity of the similarity metric is reduced. Second, words which occur in similar contexts are represented by similar feature vectors in Λ_k. Finally, noise in W are removed by simply omitting the less salient dimensions in U.

3.2.2 Applying LSA

A sentence s_i is represented by its term frequency vector f_i, where f_{ij} is the frequency of term j in s_i. Given Λ_k, the "meaning" of s_i is computed by eq. 3. Informally, s_i is represented by the sum of the LSA feature vectors. Inter-sentence similarity is estimated by the cosine of the corresponding λ (eq. 4, λ_{ik} is the $k-$th element in λ_i).

$$\lambda_i = \sum_j f_{ij} \times \Lambda_k(j) \tag{3}$$

$$M_{ij} = cos(\lambda_i, \lambda_j) = \frac{\sum_k \lambda_{ik} \times \lambda_{jk}}{\sqrt{\sum_k \lambda_{ik}^2 \times \sum_k \lambda_{jk}^2}} \tag{4}$$

3.2.3 LSA parameters

Since Λ_k is derived from the co-occurrence matrix A, the size of each training text $\delta_i \in \Delta$ is crucial to its performance. Work in information retrieval uses $\delta_i =$ *document* since the aim is to distinguish entire texts. $\delta_i =$ *paragraph* is popular in psychology experiments. However, we suspect the segmentation task may benefit from $\delta_i =$ *sentence*. Thus, two training corpora were derived from the Brown Corpus (Marcus et al., 1993). Annotations were first removed to leave a set of tokenised raw text (1.2 million tokens). This was partitioned into 35,000 paragraphs or 104,000 sentences, as two training corpora.

The parameter k adjusts the accuracy of Λ_k. A large k implies minor differences in the feature space are significant. Thus, they should be taken into account in the formulation of Λ_k. This is appropriate when the vocabulary is small and there is sufficient training data. A small k is used when A is sparse and the values in A are inaccurate.

3.3 Image ranking

Once the similarity matrix M is calculated for the input text S, the image ranking procedure in C99 is then applied to obtain a rank matrix R. R_{ij} is the proportion of neighbours of M_{ij} (11×11 grid) with a lower value than M_{ij}.

The motivation for applying image ranking in the new algorithm is to test whether a quantitative or qualitative interpretation of the similarity values has any impact on segmentation accuracy. The hypothesis is LSA similarity values are more accurate than cosine similarity values. Thus, image ranking should have a smaller impact on LSA than the cosine metric.

3.4 Clustering

The input matrix X can either be the similarity matrix M or the rank matrix R, depending on whether ranking is applied to M. Topic boundaries are identified by the divisive clustering procedure in C99. A topic segment t_k is defined by its start and end sentences, s_i and s_j, or its range $t_k = [i, j]$. The number of inter-sentence similarity values in t_k is $\alpha(t_k) = |t_k|^2$. The sum of the values in t_k is $\beta(t_k) = \sum_{i \in t_k} \sum_{j \in t_k} X_{ij}$. Thus, the average inter-sentence similarity value for a segmentation $T = \{t_1, ..., t_n\}$ is defined as,

$$\mu_T = \frac{\sum_{k=1}^n \beta(t_k)}{\sum_{k=1}^n \alpha(t_k)}$$

The divisive clustering algorithm begins by considering the entire input document S as a coherent topic segment. This is partitioned into two segments $T = \{t_1, t_2\}$ at a sentence boundary that maximises μ_T, i.e. the most prominent topic boundary. The recursive procedure proceeds until S can no longer be subdivided. The optimal segmentation is signalled by a sharp change in μ_T. For implementation details and optimisations, see (Choi, 2000a).

4 Evaluation

The following experiments aim to establish the relationship between linguistic processes (stemming, ranking, cosine metric, LSA) and segmentation error rate. The test procedure is based on that presented in (Choi, 2000a) which was derived from work in TDT (Allan et al., 1998) and previous experiments in text segmentation (Reynar, 1998, 71-73). The task is to find the most prominent topic boundaries in a concatenated text.

4.1 Experiment procedure

The accuracy of a segmentation algorithm is assessed by the experiment package[1] described in (Choi, 2000a). A test sample is a concatenation of ten text segments. Each segment is the first n sentences of a randomly selected document from a subset[2] of the Brown corpus (Marcus et al., 1993). Table 1 presents the corpus statistics. A sample is characterised by the range of n. $T_{i,j}$ is a set of samples with $i \leq n \leq j$. T is the union of the other four test sets.

	$T_{3,11}$	$T_{3,5}$	$T_{6,8}$	$T_{9,11}$	T
Samples	400	100	100	100	700

Table 1: Test corpus statistics.

Segmentation accuracy is measured by the metric proposed in (Beeferman et al., 1999). Let T_r and T_p be the reference segmentation and that proposed by an automatic procedure. k is the average segment length in T_r. $p(\text{same}|T_r, k)$ and $p(\text{diff}|T_r, k)$ refer to the likelihood of sentence s_i and s_{i+k} belonging to the same and different topic segment(s) in T_k. $p(\text{same}|T_r, T_p, \text{diff}, k)$ is the probability of a *miss*, i.e. s_i and s_{i+k} are about different topics in T_k but they belong to the same topic segment in T_p. $p(\text{diff}|T_r, T_p, \text{same}, k)$ is the probability of false alarm, i.e. two sentences are about the same topic in T_r but they belong to different segments in T_p. Equation 5 combines these four measures to calculate $p(\text{error}|T_r, T_p, k)$, the probability of segmentation errors. The error rate of an algorithm is computed as the average of $p(\text{error}|T_r, T_p, k)$ for a test set. A low error rate implies high segmentation accuracy.

$$\begin{aligned} &p(\text{error}|T_r, T_p, k) = \\ &\quad p(\text{same}|T_r, T_p, \text{diff}, k)p(\text{diff}|T_r, k) + \\ &\quad p(\text{diff}|T_r, T_p, \text{same}, k)p(\text{same}|T_r, k) \end{aligned} \tag{5}$$

This test procedure is not perfect. First, assessing the accuracy of an algorithm in an artificial task is inferior to a test that uses human segmented text. However, this approach does allow us to conduct a large-scale comparative study on similarity metrics which focuses on text similarity rather than topic boundary detection. Second, the error metric favours texts with short topic segments. Segmentation errors within a segment which is smaller than k are not always detected correctly. Thus, an algorithm is assessed using texts with different ranges of segment length. Although the metric is not perfect, it is significantly more accurate than the popular precision/recall metric which ignores near misses. Furthermore, the method is sufficiently accurate for this comparative study.

4.2 Experiment 1 – Baseline

Five degenerate algorithms define the baseline for the experiments. B_e partitions a document into $e = 10$ segments of equal length. B_n does not propose any boundaries. B_a assumes all potential boundaries are topic boundaries. B_b randomly selects $b = 10$ boundaries. $\mathrm{B}_?$ randomly selects any number of boundaries as real boundaries. Details about B_b and $\mathrm{B}_?$ are described in (Choi, 2000a).

	$T_{3,11}$	$T_{3,5}$	$T_{6,8}$	$T_{9,11}$	T
B_e	45%	38%	39%	36%	42%
B_n	46%	47%	47%	47%	47%
B_a	54%	53%	53%	53%	53%
B_b	46%	47%	47%	47%	47%
$\mathrm{B}_?$	54%	53%	53%	53%	53%

Table 2: Error rate: baseline algorithms.

Table 2 shows B_e performed best with an average error rate of 42%. This is the baseline for algorithms that find the e most prominent topic boundaries. $\mathrm{B}_?$ serves as the baseline for methods that determines the optimal segmentation, i.e. the number of topic segments in a text.

4.3 Experiment 2 – An analysis of C99

The aim is to relate stemming, ranking and the termination procedure in C99 with segmentation accuracy. The algorithm used in this experiment is identical to that presented in (Choi, 2000a) except tokens such as `--` and `-` are recognised as punctuation marks and removed during pre-processing. Test results show this modification reduces error rate by 1%. An analysis of the original algorithm reveals that non-word tokens introduce errors since they are converted into a null string by the stemming algorithm (Porter, 1980).

This implementation of C99 has three parameters. $+r$ implies ranking is applied to the similarity matrix prior to divisive clustering. $+s$ shows the stemming algorithm is used in pre-processing. $+b$ means the algorithm finds the 10 most prominent topic boundaries, i.e. the automatic termination procedure is inactive.

Test results (table 3) show ranking is crucial to C99. There is a 10% difference between row 3 and 6

[1] `http://www.cs.man.ac.uk/∼choif/software`
Package name : `C99-1.2-release.tgz`

[2] News articles `ca**.pos` and informative text `cj**.pos`.

r	s	b	$T_{3,11}$	$T_{3,5}$	$T_{6,8}$	$T_{9,11}$	T
+	+	+	12%	11%	9%	9%	11%
+	+	-	13%	17%	10%	10%	12%
+	-	+	13%	10%	10%	10%	12%
+	-	-	13%	18%	10%	12%	13%
-	+	+	21%	18%	19%	18%	20%
-	-	+	23%	19%	21%	20%	22%

Table 3: Error rate: variants of C99.

for T. This confirms the cosine metric is inaccurate for short text segments but the order between values, or rank, is significant. Future experiments will establish the relationship between segment size and accuracy.

Stemming is generally believed to improve segmentation accuracy. This is confirmed by the experiment results. However, we discovered that the process can introduce errors when segmenting short segments. There is a 0.7% difference between row 1 and 3 for $T_{3,5}$.

Finally, the termination strategy in C99 is not effective for short topic segments. There is a 6.3% improvement between row 1 and 2 for $T_{3,5}$. However, its performance for larger segments is exceptional (0.6% difference between row 1 and 2 for T).

4.4 Experiment 3 – Latent semantic analysis

The aim is to establish the relationship between LSA dimensionality, training data and accuracy. Our new algorithm, CWM, was used in this experiment. The method is identical to C99 except the stemming algorithm has been disabled and LSA is used in the formulation of the similarity matrix. Ten LSA spaces were examined. Each space is characterised by the training data and its dimensionality. s and p imply the LSA space was trained on sentences and paragraphs, respectively. $[100, 500]$ represent the dimensionality of the trained space. For instance, $(p, 400)$ is a 400-dimensional space that was trained on paragraphs. Like C99, +r implies ranking is applied to the similarity matrix. +b means CWM finds the ten most prominent boundaries.

Let μ be the column average. Test results (table 4) show ranked LSA (column 4) has the lowest error rate. The raw values (column 1 and 3) performed well. The 1% difference in accuracy implies the termination strategy works well with LSA. However, the same method is not applicable to the ranked LSA values (See column 2).

The results in column 3 highlights the relationship between LSA space and error rate. On average, a LSA space that was trained on paragraphs ($\mu(p) = 11.8\%$) out-performed one that was trained on sentences ($\mu(s) = 15.6\%$). This shows similarity is well modelled by word co-occurrence in paragraphs. It also suggests that although sentences are good for identifying words about the same topic, paragraphs are better for finding dissimilar words.

r	-	+	-	+
b	-	-	+	+
$s, 100$	16%	35%	15%	15%
$s, 200$	17%	40%	15%	13%
$s, 300$	17%	42%	16%	12%
$s, 400$	18%	43%	16%	11%
$s, 500$	18%	44%	16%	10%
$p, 100$	12%	34%	11%	10%
$p, 200$	13%	40%	11%	10%
$p, 300$	13%	41%	12%	9%
$p, 400$	14%	42%	12%	8%
$p, 500$	14%	43%	13%	8%
μ	15%	40%	14%	11%

Table 4: Error rate: LSA parameters and CWM.

Intuitively speaking, large feature vectors are expected to generate more accurate similarity values. Thus, segmentation accuracy should improve with dimensionality. The figures in column 3 show high dimensionality increases error rate. However, the figures in column 4 suggest the contrary. This implies high dimensionality improves the ranking of LSA values but is detrimental to value accuracy.

4.5 Experiment 4 – A comparative study

	$T_{3,11}$	$T_{3,5}$	$T_{6,8}$	$T_{9,11}$	T
$\text{CWM}_{(500,r,b)}$	9%	10%	7%	5%	8%
$\text{U00}_{(b)}$	10%	9%	7%	5%	9%
$\text{CWM}_{(100,b)}$	12%	10%	9%	8%	11%
$\text{C99}_{(s,r,b)}$	12%	11%	9%	9%	11%
$\text{U00}_{(?)}$	12%	9%	10%	11%	11%
$\text{C99}_{(s,r)}$	13%	17%	10%	10%	12%
$\text{CWM}_{(500,b)}$	14%	10%	11%	12%	13%
$\text{C99}_{(b)}$	23%	19%	21%	20%	22%

Table 5: Error rate: a comparative study.

Table 5 presents a summary of experiment results. All variants of CWM uses a LSA space that was trained on paragraphs. $\text{CWM}_{(500,r,b)}$ is the new algorithm that uses Λ_{500} for similarity estimates. $\text{CWM}_{(500,b)}$ is the same algorithm except ranking has been disabled. $\text{CWM}_{(100,b)}$ uses Λ_{100}. $\text{C99}_{(s,r,b)}$ is the same as $\text{CWM}_{(100,b)}$ and $\text{CWM}_{(500,b)}$, except stemming is applied during pre-processing and it uses the cosine metric to measure similarity. U00 is the method proposed in (Utiyama and Isahara, 2001).

Test results show $\text{CWM}_{(500,r,b)}$ is more accurate than previous algorithms. The two-fold increase in accuracy between $\text{CWM}_{(100,b)}$ and $\text{C99}_{(b)}$ implies

LSA is a more accurate similarity measure than the cosine metric. Finally, the difference between $\mathrm{CWM}_{(500,r,b)}$ and $\mathrm{CWM}_{(500,b)}$ shows ranking improves segmentation accuracy. The significance of our results has been confirmed by both t-test and KS-test (Press et al., 1992).

5 Conclusions

A series of experiments were conducted to establish the relationship between linguistic processes and segmentation accuracy. C99 (Choi, 2000a) was used as the test bench. In the first set of experiments, its stemming algorithm, ranking procedure and automatic termination method were systematically disabled to determine the contribution of each process to overall performance. We discovered that, first, stemming generally improves accuracy unless the topic segments are short (3 to 5 sentences). Second, ranking plays a vital role in C99. It reduces error rate by half (22% to 10%). Finally, the termination procedure in C99 is effective (0.6% difference). The method works particularly well on long topic segments (> 6 sentences).

The second set of experiments focused on LSA as a similarity metric. The cosine metric in C99 was replaced by LSA. Ten different LSA spaces were examined. We discovered that LSA is twice as accurate as the cosine metric. The results also showed vocabulary difference between paragraphs is a good feature for training a similarity metric. Further investigation into the relationship between ranking, LSA dimensionality and error rate revealed that LSA values become less accurate as more dimensions are incorporated into the feature vectors. This implies the training data is noisy. However, with ranking, error rate decreases. This shows the order of LSA values becomes more accurate when more features are used.

Future work will focus on document specific LSA and the termination strategy of the new algorithm. Test results have shown the termination procedure in C99 works well on LSA similarity values but not on the ranked values. We suspect the threshold selection method has to be modified. In terms of clustering, dynamic programming approaches (Ponte and Croft, 1997; Utiyama and Isahara, 2001, for example) will be examined. Finally, a LSA procedure for computing document specific similarity values will be evaluated.

References

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop.*

Doug Beeferman, Adam Berger, and John Lafferty. 1997. Text segmentation using exponential models. In *Proceedings of EMNLP-2.*

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning, special issue on Natural Language Processing*, 34(1-3):177–210. C. Cardie and R. Mooney (editors).

W. Chafe. 1979. The flow of thought and the flow of language. In T. Givon, editor, *Syntax and Semantics: Discourse and Syntax*, pages 159–182. Academic Press.

Freddy Y. Y. Choi. 2000a. Advances in domain independent linear text segmentation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 26–33, Seattle, USA, May. ACL.

Freddy Y. Y. Choi. 2000b. Improving the efficiency of speech interfaces for text navigation. In *Proceedings of ICCHP'00.*

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

David Eichmann, Miguel Ruiz, and Padmini Srinivasan. 1999. A cluster-based approach to tracking, detection and segmentation of broadcast news. In *Proceedings of the 1999 DARPA Broadcast News Workshop (TDT-2).*

G. H. Golub and C. F. van Loan. 1989. *Matrix Computations.* John Hopkins University Press.

Mochizuki Hajime, Honda Takeo, and Okumura Manabu. 1998. Text segmentation with multiple surface linguistic cues. In *Proceedings of COLING-ACL'98*, pages 881–885.

Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English.* Longman Group, New York.

Marti Hearst and Christian Plaunt. 1993. Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, Pittsburgh, PA.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the ACL'94.* Las Crces, NM.

Oskari Heinonen. 1998. Optimal multi-paragraph text segmentation by dynamic programming. In *Proceedings of COLING-ACL'98.*

Y. Jing and W. B. Croft. 1994. An association thesaurus for information retrieval. In *Proceedings of RIAO'94, Intelligent Multimedia Information Retrieval Systems and Management*, pages 285–298.

Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. 1998. Linear segmentation and segment significance. In *Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6)*, pages 197–205, Montreal, Quebec, Canada, August.

M. Kaszkiel and J. Zobel. 1997. Passage retrieval revisited. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185, Philadelphia. ACM.

Stefan Kaufmann. 1999. Cohesion and collocation: Using context vectors in text segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (Student Session)*, pages 591–595, College Park, USA, June. ACL.

D. Kieras. 1982. A model of reader strategy for abstracting main ideas from simple technical prose. *Text*, 2(13).

K. Koskenniemi. 1983. *Two-level Morphology: a General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, Department of General Linguistics, University of Helsinki.

Hideki Kozima. 1993. Text segmentation based on similarity between words. In *Proceedings of ACL'93*, pages 286–288, Ohio.

Sadao Kurohashi and Makoto Nagao. 1994. Automatic detection of discourse structure by checking surface information in sentences. In *Processings of COLING'94*, volume 2, pages 1123–1127.

T.K. Landauer and S.T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Diane J. Litman and Rebecca J. Passonneau. 1995. Combining multiple knowledge sources for discourse segmentation. In *Proceedings of the 33rd Annual Meeting of the ACL*.

R. Longacre. 1979. The paragraph as a grammatical unit. In T. Givon, editor, *Syntax and Semantics: Discourse and Syntax*, pages 115–134. Academic Press.

Mitch Marcus, M. A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2).

S. Miike, E. Itoh, K. Ono, and K. Sumita. 1994. A full text retrieval system. In *Proceedings of SIGIR'94*, Dublin, Ireland.

J. Morris and G. Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, (17):21–48.

Jane Morris. 1988. Lexical cohesion, the thesaurus, and the structure of text. Technical Report CSRI 219, Computer Systems Research Institute, University of Toronto.

Jay M. Ponte and Bruce W. Croft. 1997. Text segmentation by topic. In *Proceedings of the first European Conference on research and advanced technology for digital libraries*. U.Mass. Computer Science Technical Report TR97-18.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137, July.

William H. Press, Saul A. Teukolsky, William T. Vettering, and Brian P. Flannery, 1992. *Numerical recipes in C: The Art of Scientific Computing*, chapter 14, pages 623–628. Cambridge University Press, second edition.

Jeffrey C. Reynar. 1994. An automatic method of finding topic boundaries. In *Proceedings of ACL'94 (Student session)*.

Jeffrey C. Reynar. 1998. *Topic segmentation: Algorithms and applications*. Ph.D. thesis, Computer and Information Science, University of Pennsylvania.

Jeffrey C. Reynar. 1999. Statistical models for topic segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 357–364, College Park, USA, June. ACL.

Gerard Salton, James Allan, and Chris Buckley. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, pages 49–58, Pittsburgh, PA.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proceedings of ACL'2001*, Toulouse, France, July. To appear.

C. J. van Rijsbergen. 1979. *Information Retrieval.* Buttersworth.

P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. 1999. Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In S. Lajoie and M. Vivet, editors, *Artificial Intelligence in Education*, pages 535–542, Amsterdam. IOS Press.

M. Wolfe, M. E. Schreiner, B. Rehder, D. Laham, P. W. Foltz, W. Kintsch, and T. K. Landauer. 1998. Learning from text: Matching readers and texts by Latent Semantic Analysis. *Discourse Processes*, 25:309–336.

Jinxi Xu and Bruce W. Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, Zurich, Switzerland, August.

Yaakov Yaari. 1997. Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of RANLP'97*. Bulgaria.

Gilbert Youmans. 1991. A new tool for discourse analysis: The vocabulary-management profile. *Language*, pages 763–789.