Computational Intelligence, Volume 16, Number 4, 2000

SECOND-ORDER COHESION

STEFAN KAUFMANN
Linguistics Department and Center for the Study of Language and Information, Stanford University, Stanford, CA 94305, USA

Similarity in contextual behavior between words is considered a source of "lexical cohesion," which is otherwise hard to measure or quantify. Such contextual similarity is used by an implementation for text segmentation, the `VecTile` system, which uses precompiled vector representations of words to produce similarity curves over texts. The performance of this system is shown to improve over that of the `TextTiling` algorithm of Hearst (1997).

Key words: text segmentation, information retrieval, cohesion, singular value decomposition.

1. INTRODUCTION

The notion of text cohesion rests on the intuition that a text is held together by a variety of internal forces. Hoey (1991) defines it as follows:

> Cohesion is a property of text whereby certain grammatical or lexical features of the sentences of the text connect them to other sentences in the text [p. 266].

The generality of this definition reveals a fundamental problem with the concept: Granting that there is such a thing as cohesion in language, it is likely not due to some one linguistic feature but emerges instead from the combined effect of a variety of only loosely related forces. Under this assumption, the ultimate questions to be answered by the ongoing effort to understand cohesion are

- What are the factors contributing to cohesion?
- Is there a robust correlation between cohesion and measurable properties of the text?
- Can the contribution of the various factors be quantified and weighed in relative or absolute terms?

The first of these questions has been paid much attention for over 20 years, and the general consensus is briefly reviewed in this section. Some of the phenomena involved are better understood than others.

A procedure to evaluate the performance of text segmentation systems (comparison with subject judgments, here discussed and performed in Section 4) is generally believed to help answer the second question, although in an indirect way. The problems with it are discussed in Section 2.1.

The third question has been given little, if any, attention in the literature, which is not surprising, given that a consensus on the second is only beginning to emerge.

1.1. The Linguistic Notion of Cohesion

Much of the linguistic work on text cohesion is indebted to Halliday and Hasan (1976), and it is instructive to start by reviewing their account before discussing how the work presented here fits in that general framework.

Address correspondence to the author at the Linguistics Department, Stanford University, Stanford, CA 94305; e-mail: kaufmann@csli.stanford.edu.

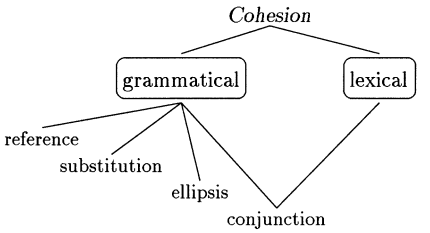

FIGURE 1. Halliday and Hasan's forms of cohesion.

A text, Halliday and Hasan argue, is not a structurally definable unit of language in the same way as sentences or parts of sentences are:

> A text is not something that is like a sentence, only bigger; it is something that differs from a sentence in kind [p. 2].

Thus, if cohesion is not a structural property, what is it then?

1.1.1. Grammatical Cohesion. Halliday and Hasan define *cohesion* as a network of relationships between locations in the text, both intrasentential and intersentential. The most obvious examples are cases where the interpretation of some expression is determined by, or even depends on, the presence of an expression elsewhere. A natural example of this is *coreference* between anaphoric, cataphoric, and "exophoric" (deictic) items and their antecedents. Other major classes identified by Halliday and Hasan are *substitution* (e.g., *one* for nouns, *do so* for verb phrases, and *it* for clauses—the distinction between this class and the preceding one is subtle), *ellipsis*, and *conjunction* [sentential connectives of various kinds, including conjunctions in the logical sense, but also items specialized on indicating causal or temporal relations (p. 242)]. The way these groups are dealt with in the book is sketched in Figure 1.

While drawing a distinction between *grammatical* cohesion, on the one hand, and *lexical* cohesion, on the other, Halliday and Hasan emphasize that the two are not fundamentally different in nature, but the division is nevertheless of practical importance because they are realized by quite different means: All cases of *grammatical* cohesion have in common that one element in the discourse depends on the presence of another and that this dependence is made obvious by the choice of words (pro-forms, etc.) While it is generally hard to quantify cohesion, the grammatical cases are still relatively easy to analyze.

The same is not true of lexical cohesion. As Halliday and Hasan note, the latter does not rely on structural clues, it can work over longer distances, and its absence does not render the text unintelligible. Accordingly, one might conclude that of all the varieties of cohesion, it is the one that neither depends on nor facilitates grammatical analysis.

1.1.2. Lexical Cohesion. Lexical cohesion arises from the mere presence in the text of lexical items that "hang together" by virtue of their meaning and their vicinity. The challenge they present to the linguistic analyst lies in the difficulty of detecting and measuring the degree of semantic "relatedness" given any two words.

Halliday and Hasan's (1976) discussion of lexical cohesion deals with two varieties: *reiteration* and *collocation*. The difference between the two is that while the former preserves reference between two related items, the latter does not. The former is realized on a continuum from simple repetition of the same word to use of a related but more

general word with the same referent. This is exemplified by sentences such as those in Example 1:

Example 1. There's a boy climbing the old elm.

a. *That elm* isn't very safe.
b. *That tree* isn't very safe.
c. *That old thing* isn't very safe.

(Halliday and Hasan 1976, p. 280; emphasis added)

Here, the relationships of the various nouns to the plant can be defined in terms of a hierarchical ontology in which more general terms occupy higher positions, making the sentences in Example 1 a procession from the specific to the general. Later work in artificial intelligence has taken up this intuition and attempted to use thesauri to find connections of that kind (see Section 2.1.) Furthermore, the coreference is manifest structurally in the use of the definite article.

It is not generally the case, however, as Halliday and Hasan recognize, that the items in such a relationship have the same referent. In fact, they need not even stand in any easily identifiable relationship. Cases of this kind are called *collocations*. An example is

Example 2. a. Why does this little *boy* wriggle all the time?
b. *Girls* don't wriggle. (p. 285)

Words enter collocational relationships primarily because they "regularly co-occur" (p. 284). This is the defining criterion, while the question as to what precisely the *semantic* relationship between them is, unlike in the case of reiterations, is not particularly helpful in the analysis. The reason for this is that words in virtually any conceivable semantic relationship, including complementarity as in Example 2, but just as easily near-synonyms (*ascend . . . climb*), cohyponyms (*chair . . . table*), antonyms (*cold . . . hot*), words taken from the same series (*Monday . . . Friday*), etc., when taken together, are capable of contributing to cohesion.[1] The only generalization that can be made, then, is that there is a correlation between occurrence in the same contexts and cohesion-forming association.

The strong dependence of the collocational behavior of a word on the particular text and the immediate context of every instance is stressed by Halliday and Hasan in a way that resembles what will be presented below:

> Without our being aware of it, each occurrence of a lexical item carries with it its own textual history, a particular collocational environment that has been built up in the course of the creation of the text and that will provide the context within which the item will be incarnated on this particular occasion [p. 289].

The method to be introduced below, like others found in the computational literature, is intended to capture such environments of individual instances of a word to build a "profile" of that word in use, and to use that profile, rather than the word itself, in estimating cohesion.

Further, Halliday and Hasan discuss two notions of "relatedness," one being co-occurrence in the text and the other being "degrees of proximity in the lexical system, a function of the probability with which one tends to co-occur with another" (p. 290). In other words, much knowledge about words can be gleaned from "the company they keep." Below, these degrees of proximity are implemented as angles between vectors in a high-dimensional space.

[1]For more examples, see Halliday and Hasan (1976, pp. 284–288).

2. COMPUTATION

We saw above in Section 1 that lexical cohesion neither facilitates nor lends itself easily to semantic analysis. From a computational point of view, it is precisely these properties that make it a promising object of study for two reasons. First, with other sources of cohesion, especially reference, substitution, and ellipsis, the dependencies have to be found first in order to establish whether they contribute to cohesion. This is not a problem for the linguist, but for a computer, this enabling task tends to be difficult and error-prone. Detecting lexical cohesion, on the other hand, does not require a thorough understanding of the text but only a suitable representation of the vocabulary.

Second, the kind of knowledge that may help to facilitate the treatment of lexical cohesion, which, as Halliday and Hasan (1976) note, ought to be obtained through statistical observations on occurrence patterns, may be acquired and used computationally. In short, detecting and measuring lexical cohesion are the kinds of tasks that are hard for humans to perform consciously but feasible for computers.

2.1. Assumptions and Problems

A pervasive problem with algorithms intended to measure cohesion is their evaluation. Since cohesion itself is an elusive concept, what would be its measurable indicator? An analogue in physics and material science would be *tensile strength*, measured by stretching an object until it breaks.

There is no equivalent for that kind of experiment in linguistics. Instead, the algorithms typically calculate a measure of *similarity* between parts of the text. How the measure is obtained varies widely, but the common assumption is that a measure of similarity, obtained in the absence of full semantic understanding and reasoning, must have to do with cohesion. If, furthermore, *grammatical* cohesion is discounted by ignoring anaphoric links, pro-forms, and the like, only *lexical* cohesion remains. The similarity measure is then assumed to correlate closely with, indeed to be a measure of, lexical cohesion.

In the computational literature on the subject, the preceding assumption is usually made implicitly. Then a high similarity measure between adjacent pieces of text is assumed to indicate that there are strong cohesive forces across the gap between the two. The evaluation of the `VecTile` system below follows this convention, but it should be kept in mind that what is measured is, strictly speaking, not cohesion itself.

A further problem is that it is hard to know whether the obtained measure of similarity is good. The only way to answer this question is to use it in a system performing some practical task, such as text segmentation, and evaluating the performance of the system. This is not without problems, since cohesion is likely not the only factor responsible for subjects' decisions.

Thus there are two major problems with the evaluation: (1) The system can be said to measure cohesion only under the assumption that whatever is observed is an indicator of it, and (2) cohesion is likely to be only one of many factors determining subjects' judgments.

2.2. Previous Approaches

Computational systems designed to calculate cohesion differ in the kinds of lexical relationships they quantify and in the amount of semantic knowledge they rely on. *Topic parsing* (Hahn 1990) uses both grammatical cues and semantic inference based on

precoded domain-specific knowledge. More general approaches assess word similarity based on thesauri, as in the *lexical chains* approach (Morris and Hirst 1991) or dictionary definitions (Kozima 1993).

Methods that solely use observations of patterns in vocabulary use include *vocabulary management* (Youmans 1991) and the *blocks* algorithm implemented in the `TextTiling` system (Hearst 1997). The latter is compared below with the system introduced here, and more will be said about it below. Although there are previous approaches based on vector spaces (Salton and Allen 1993; Salton et al. 1996), these are only remotely related to the one discussed here.

An extensive recent overview of these and a few other approaches can be found in Chapters 4 and 5 of Reynar (1998).

3. SECOND-ORDER SIMILARITY

The method used in the present version of `VecTile` is based on the WordSpace model of Schütze (1997, 1998). The idea is to represent words by encoding the environments in which they typically occur in texts. Such a representation can be obtained automatically and often provides sufficient information to make deep linguistic analysis unnecessary. This has led to promising results in information-retrieval and related areas.

Let a dictionary W and a relatively small set C of meaningful "content" words—likely, although not necessarily, a subset of W—be given. Then, for each word $w \in W$ and each $c \in C$, the number of times is recorded that the two co-occur in some text corpus, where co-occurrence can be within the same document or some other unit (in the present implementation, it was a window of 35 words). This yields a number for every member of $C \times W$ or, put differently, a $|C|$-dimensional vector for each $w \in W$. The direction that the vector has in the resulting $|C|$-dimensional space can then be thought of as a representation of the collocational behavior of the word in the training corpus.

In the present implementation, $|W| = 20{,}500$ and $|C| = 1000$. This is still too large to handle efficiently on most computers, and besides, depending on the corpus, for many pairs in $C \times W$ the number of co-occurrences may be zero. For these reasons, the matrix of $20{,}500 \times 1000$ values is reduced to one of $20{,}500 \times 100$ (possibly negative) values using singular-value decomposition (SVD; see Golub and van Loan, 1989), an operation that relocates the vectors in the lower-dimensional space, approximating their mutual distances in the high-dimensional space as well as possible.[2]

The vector representations of the words can then be manipulated in various ways. As a measure of similarity in collocational behavior between two words, the cosine between their vectors is computed: Given two n-dimensional vectors $\vec{v}$, $\vec{w}$,

$$\cos(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2 \sum_{i=1}^{n} w_i^2}}$$

The cosine is a number between 0 and 1, indicating that the vectors are orthogonal (for the former), have the same direction (for the latter), or stand in some intermediate angle. Large cosine values correspond to small angles.

[2]The use of SVD in this domain was first suggested for latent semantic indexing (LSI), a related method in information retrieval in which matrices of document-word co-occurrences, rather than word-word co-occurrences, are treated in a similar way in order to obtain vector representations of documents (Deerwester et al. 1990).

In order to represent larger pieces of text than words, one can add up the vectors of the constituent words. In this way, adding word vectors yields new vectors in the same space, which can again be compared against each other and word vectors. One can then, for example, add up the words in two adjacent portions of text and calculate the cosine between the two resulting vectors to get a measure of the lexical similarity between the two portions. This is what the `VecTile` algorithm does.

Thus the comparison between two words w and v is in a sense indirect, mediated by the dimension labels, fixed landmarks in the the concept space. This makes it different from similarity measures that depend on direct reference to the co-occurrence (w, v) between two words w and v.

Lee (1999) and Dent and Mercer (1999) discuss information-theoretic similarity measures based on the relative frequency of the pair (w, v) or on the conditional probability of w given v. There are differences between such approaches and the present one that merit deeper investigation; in the present context, no more than a glance at them can be offered.

One such difference of some practical value is this: In the former, co-occurrence enters the calculation for each pair. Given two words w and v, the numbers associated with each of them are useless unless they are supplemented with counts on the pair (w, v). In the system presented here, on the other hand, words have independent representations, allowing for any two of them to be compared directly with a very simple algorithm. Indeed, two words may receive a high score without ever co-occurring, as long as they do frequently co-occur with the same content-bearing words.

Another difference is that, as mentioned earlier, the similarity between two words is always "indirect," mediated by the column labels. The more columns there are, the more reliable the similarity score is. Therefore, the measure of similarity obtained from two word vectors, whose dimensions are labeled with words, cannot be decomposed in any useful sense into individual scores based on those dimensions' labels. The words that were used to calculate the word vectors become uninteresting. It is in this sense that the term *second order* is used here.

A third difference is the dependence of the WordSpace model on the choice of the dimension labels. Where they are themselves situated in the concept space heavily influences the measure of similarity between words. This is usually not the case in information-theoretic accounts, where given two words v and w, only the frequencies $f(v)$, $f(w)$, and $f(vw)$ and the size of the corpus, but not the relative frequency with respect to words other than v or w, enter the calculation.

3.1. Word Similarities

The main rationale for considering the WordSpace approach is the hypothesis that

- Words with similar meanings tend to be used in similar contexts, coupled with the fact that
- Words used in similar contexts tend to have vectors in similar directions.

Under these assumptions, then, it follows that a large cosine value between vectors indicates semantic similarity. Evidence that this inference holds has been offered in various places (e.g., Schütze 1997, pp. 90–91).

To illustrate, Table 1 lists the words closest to the example word *chair*, where the vectors are calculated using two different corpora: from the New York Times Newswire, written around 1994–1995 ("NYT") and from the Associated Press Newswire, written

TABLE 1. Nearest Neighbors Ordered by Vector Similarity

NYT		AP	
chairs	0.612997	row	0.566109
floor	0.604490	execution	0.551683
sitting	0.586961	inmate	0.530925
room	0.573644	murderer	0.520940
eyes	0.562481	bundys	0.509287
door	0.559803	death	0.475072
stared	0.555688	alabamas	0.468611
table	0.546852	convicts	0.461459
bathroom	0.542431	convict	0.453665
glass	0.535583	stabbing	0.443597

around 1988–1989 ("AP").[3] Note that the semantic relationships between the words, although obviously present, cannot be uniformly characterized. In general, these associations have very much the flavor of the collocations discussed in Halliday and Hasan (1976).

The table also shows where this approach can be vulnerable: In the vector space based on the AP Newswire, all the words related to *chair* are related to capital punishment. This is not likely to be the association most speakers would think of first. The example illustrates the strong dependence of the derived vector space on the training corpus. In information retrieval, this feature can be taken advantage of in order to build personalized or domain-specific search engines. It also may influence the performance of text segmentation systems (see below.)

3.2. The `VecTile` Algorithm (Figure 2)

This section gives an outline of the segmentation algorithm implemented in the system and the parameter settings with which it was run in the experiment described below.

Input. The document to be segmented is read in plain text form, with paragraph breaks preserved.

Gap Values. The main function of the algorithm is to use the word vectors to calculate similarity scores between adjacent blocks of text. Two adjacent windows are moved over the text, and the similarity between them is calculated at given intervals. The score is obtained by associating every word in each window with its vector, summing the vectors for all words in the window, and calculating the cosine between the two resulting window vectors. Two important parameters involved in this operation are (1) the size of the blocks and (2) the "blockstep" *bs* (gap values are calculated only at every *bs*th input word). Those in between are counted but do not receive their own gap values. In the experiments below, the values were 200 and 10, respectively.

[3]The system has a publicly accessible interface at `http://matsu.stanford.edu/cgi-bin/semlab/webif/webdemo`.

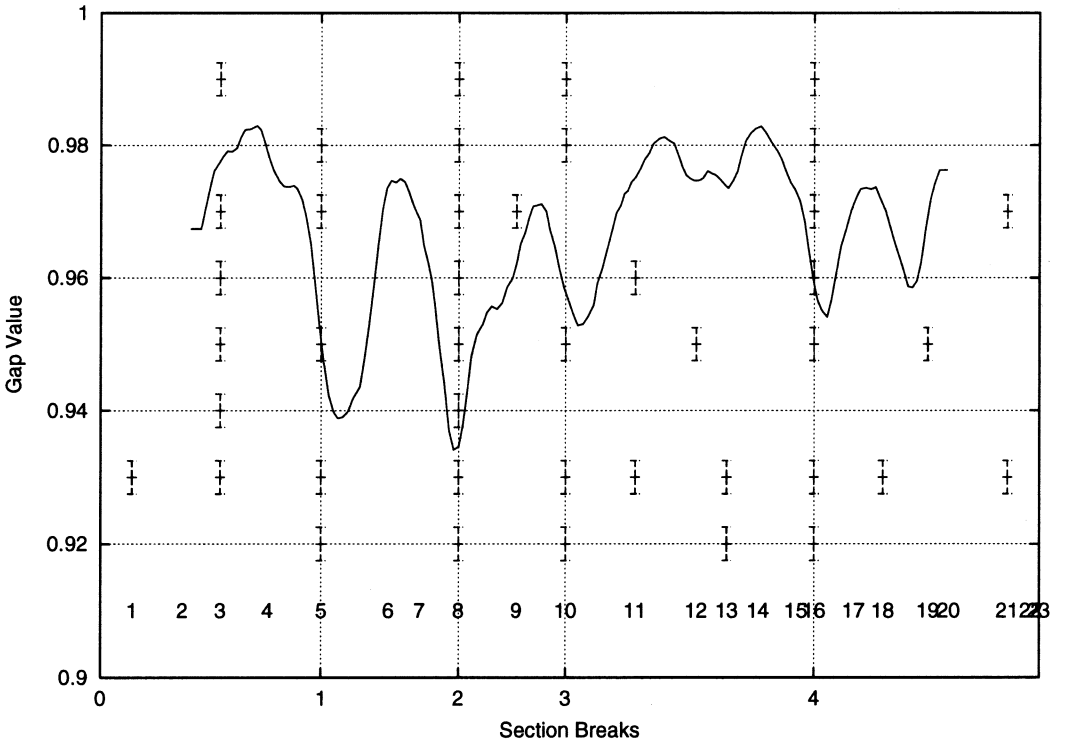

FIGURE 2. Example of a `VecTile` similarity plot (text 2). Numbers inside the plot indicate paragraphs. The judgments of eight subjects are shown as bars in eight rows.

Smoothing. The values on the gaplist represent the similarity graph over the length of the text. The troughs in the graph eventually will be judged to mark spots with little cohesion. Since the curve is usually rugged, a low-pass filter is applied to the list. The one used here is a crude device inspired by Hearst (1997). It assigns to each gap the mean of it and the gaps surrounding it. Important parameters are the size of the window used in the smoothing (3) and the number of iterations the filter takes over the gaplist (1).

Ranking. Only the deepest valleys in the graph are considered as candidates for break assignment. To make this decision depend only on the local values around it, rather than on some absolute or global value, elements of the smoothed gaplist are assigned *depth scores*. Only a subset of the gapvalues, namely the bottom points of troughs, receive depth scores. Each such point is assigned the sum of the differences between it and the highest adjacent peaks on both sides. In this way, the deepest valleys in the curve are assigned the highest depth scores. The method currently employed simply hill-climbs in both directions. This may be a bit too simple-minded and is potentially vulnerable to local maxima on a larger slope.

Selection. Of all the assigned depth scores, only the largest ones qualify for section break assignment. A preset cutoff value for the selection is problematic, whether it determines the number of breaks or some particular depth value. Instead, the cutoff is calculated as the sum of the mean of all depth values and the product of the standard deviation with the CUTFACTOR, one of the parameters of the algorithm (0.5).

Output. The break points found this way can then be associated with paragraph breaks. For each member of the gaplist that has been chosen as a break point, the nearest paragraph boundary, counting the separating words, is chosen. These locations are the output of the program.

3.3. TextTiling

The algorithm outlined in the preceding section is largely similar to the system presented by Hearst (1997). It, too, uses sliding windows and assigns scores to the gap between them, and its method of selecting section breaks from the gap values is almost identical to VecTile's. The crucial difference is that TextTiling builds window vectors solely by counting the occurrences of strings in the windows. Repetition is rewarded by the present approach, too, since identical words contribute most to the similarity between the block vectors. However, similarity scores can be high even in the absence of pure string repetition, as long as the adjacent windows contain words that co-occur frequently in the training corpus. Thus what a direct comparison between the systems will show is whether the addition of collocational information gleaned from the training corpus sharpens or blunts the judgment. How and to what extent this additional information affects the results is one of the motivating questions behind the work reported here. The other, closely related but not identical, question is whether the VecTile system affords a considerable improvement in performance.

For comparison, the TextTiling algorithm was implemented and run with the same parameter settings.

4. EVALUATION

Text segmentation is a task for which lexical cohesion is commonly considered relevant (Kozima 1993; Benbrahim and Ahmad 1995; Salton et al. 1996; Green 1997; Hearst 1997). It can serve as the enabling technique for many practical applications such as text summarization, passage retrieval, or hypertext linking. Here, too, the motivating assumption is that with text-internal topic shifts, the vocabulary will change, and by detecting such changes in the vocabulary, one should be able to infer the section breaks.

Two analyses of the same experimental data are presented here. The first follows a commonly used protocol for the evaluation of text-segmenting systems. Its results are less useful in this context because it requires decisions that introduce additional errors, namely, the choice of a "correct" solution based on speaker judgments. The second analysis takes a somewhat different look at the data without imposing a "correct" solution.

4.1. Text Segmentation

In a pilot study, eight subjects were presented with five texts from a popular-science magazine, all between 2000 and 3400 words, or between 20 and 35 paragraphs, in length. Section headings and any other clues were removed from the layout. Paragraph breaks were left in place. Thus the task was not to find paragraph breaks, but breaks between multiparagraph passages that according to the the subject's judgment marked topic

shifts. Subjects were asked to indicate these locations by placing marks in the text. All subjects were native speakers of English.[4]

4.2. Results

4.2.1. Intersubject Agreement. The level of agreement among the judges is at best at the low end of what can be considered significant. To show this, the *kappa coefficient* K, proposed by Carletta (1996), can be used. This statistic measures the amount of agreement between two subjects, relative to the agreement that would be expected by chance alone. All the paragraph boundaries can be thought of as a set of binary random variables over which the decisions of the subjects, both positive ("boundary") and negative ("nonboundary"), are distributed. The kappa coefficient takes into consideration to what extent the decisions would be expected to agree if they were evenly distributed. Carletta's (1996) formula is

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the proportion of actual agreement and $P(E)$ is the expected proportion of agreement by chance. Intuitively, then, this amounts to saying that

$$K = \frac{\text{unexpected agreement}}{\text{expected disagreement}}$$

$K > 0$ if the agreement is greater than expected by chance, and $K = 1$ if it is total. Carletta (1996) states that in other scientific communities, researchers "think of $K > 0.8$ as good reliability, with $0.67 < K < 0.8$ allowing tentative conclusions to be drawn" but cautions that similar conventions have not been established for discourse segmentation and may be a bit more problematic (p. 252).

In order to have a "correct" opinion to compare each subject to, those locations were counted as boundaries that at least four of the eight subjects had agreed on. Three of seven (Litman and Passonneau 1995; Hearst 1997) or 30% (Kozima 1993) are also sometimes deemed sufficient. Then each subject was compared with that "expert" opinion, and the mean of the four resulting Kappa values was calculated.[5] The values for the four subjects and the mean are given in Table 2.

Hearst (1997) cites related work in discourse segmentation with comparable findings. Her $\overline{K}$ with seven judges was 0.647 (p. 54). The value of the kappa measure for this particular task is sometimes questioned (for a comparison with other measures, see Heinonen, 1999), but currently, it is the measure most generally employed. To summarize, the results reported here should be considered preliminary.

4.2.2. Evaluation. Two popular measures for assessing quality in such tasks are precision and recall. They apply whenever some subset is chosen from a large set of possibilities and it is known in advance which subset would be the correct one. In the current context, precision expresses how many of the marked boundaries are correct, while recall expresses how many of the correct boundaries are marked. The former

[4]The instructions read as follows: "You will be given five magazine articles of roughly equal length with section breaks removed. Please mark the places where the topic seems to change (draw a line between paragraphs). Read at normal speed; do not take much longer than you normally would. But do feel free to go back and reconsider your decisions (even change your markings) as you go along. Also, for each section, suggest a headline of a few words that captures its main content. If you find it hard to decide between two places, mark both, giving preference to one and indicating that the other was a close rival."

[5]The kappa values were calculated using the SAS system.

TABLE 2. Kappa Values for the Eight Subjects

	Subject								
Text	1	2	3	4	5	6	7	8	$\overline{K}$
1	0.494	0.609	0.526	0.512	0.614	0.807	0.675	0.746	*0.623*
2	0.637	0.621	0.421	0.673	0.488	0.771	0.879	0.879	*0.671*
3	0.655	0.358	0.435	0.224	0.350	0.693	0.606	0.581	*0.488*
4	0.677	0.622	0.453	0.320	0.677	0.799	0.645	0.645	*0.605*
5	0.592	0.840	0.465	0.551	0.592	0.519	0.551	0.519	*0.579*
All texts	0.610	0.612	0.466	0.447	0.550	0.711	0.666	0.661	*0.590*

is high if there are few "false alarms" (false positives) and the latter if there are few "misses" (false negatives).

$$\begin{aligned}\text{Precision} &= \frac{\text{correctly marked breaks}}{\text{marked breaks}} \\ &= \frac{\text{true positives}}{\text{true positives + false positives}} \\ \text{Recall} &= \frac{\text{correctly marked breaks}}{\text{correct breaks}} \\ &= \frac{\text{true positives}}{\text{true positives + false negatives}}\end{aligned}$$

The results of the current study for `TextTiling`, `VecTile`, and the average of the eight subjects are shown in Table 3. The figures show that `VecTile` performed somewhat better than `TextTiling` on most, although not all, texts.

TABLE 3. Precision and Recall for Five Texts (in %)

	`TextTiling`		`VecTile`		Subjects	
Text	Prec	Rec	Prec	Rec	Prec	Rec
1	50	43	50	43	78	80
2	14	20	75	60	84	78
3	50	50	60	50	70	65
4	25	40	29	40	67	75
5	20	29	25	14	80	64
Avg	32	36	48	41	76	72

4.3. Boundary Strength

The method discussed so far is vulnerable to problems not directly stemming from the task of estimating cohesion. In particular, the need to make yes–no decisions as to the locations of boundaries may introduce additional errors that compromise the

TABLE 4. Correlation Coefficients between scores and subject markings

	r	signif	r^2
`TextTile`	−0.251	0.01	0.06
`VecTile`	−0.453	0.0001	0.21

results. In order to circumvent this problem, in a second evaluation, the cosine values as a gradient measure of the *strength* of boundaries were related to the subject judgments.

4.4. Results

If subjects' decisions on section boundaries and the similarity measures obtained in the segmentation systems are both related to the same patterns in the text, they are expected to be correlated. To verify this, the Pearson correlation coefficient was calculated. (This may not be completely adequate for the number of subjects, which is not continuous.)

The correlation coefficient is a number between −1 and 1. It equals zero if the variables are not correlated at all. Both −1 and 1 indicate perfect correlation. In the present case, the two variables are, for each paragraph boundary, the score assigned to that boundary and the number of subjects who judged that boundary to be a section break. Then r is expected to be negative, assuming that low scores pattern with high numbers of subject markings, and vice versa.

Table 4 shows that there was indeed a negative correlation and that it was much stronger in the case of `VecTile`. The significance level (the probability with which such a correlation could have been obtained by chance) is given in the third column. This result is encouraging, even though r^2, commonly considered a measure of the extent to which the variation in one variable can be explained by variation on the other, is quite low in both cases.

5. DISCUSSION AND FURTHER WORK

The purpose of this study was to determine whether one particular kind of lexical information, namely, co-occurrence patterns of words obtained from a large corpus, is useful as an indicator for lexical cohesion. In the taxonomy of types of cohesion, the word vectors encode only the kind that arises from collocational behavior. Other sources were ignored, but they would surely have to be part of a system intended to capture all cohesion.

As discussed in Section 2.1, the experimental setup relies on the commonly made additional assumptions that lexical cohesion can be measured indirectly by virtue of its negative correlation with topic shifts and that the decisions of speakers in segmenting text provide a closely related kind of information. Under these assumptions, the results indicate that there is a strong correlation between shifts in the vocabulary and speakers' decisions in the segmentation task.

Some factors work against the context vector method. For instance, the system currently has no mechanism to handle words for which it has no context vectors. Often it

is precisely the co-occurrence of uncommon words not in the training corpus (personal names, rare terminology, etc.) that ties text together. Such cases pose no challenge to the string-based `TextTile` system, but `VecTile` cannot use them. The best solution may be a hybrid system with a backup procedure for unknown words. The most straightforward and simple incarnation of such a hybrid system would calculate both scores at all times and combine them using an empirically chosen weighting scheme.

Another potentially important parameter is the nature of the training corpus. In the present case, it consisted mainly of news texts, while the texts in the experiment were scientific expository texts. This difference in genre, which came about merely for practical purposes of corpus availability, may have been great enough to introduce considerable noise, and a follow-up study should use the same genre for training and evaluation.

The study reported here was performed on a very small set of texts. Even within this small sample, there is considerable variation in the performance of the two systems. A larger-scale experiment with more texts would be desirable. Unfortunately, subject judgments cannot be obtained easily for more than a few texts. The experiments reported here were designed to facilitate a direct comparison with Hearst's system as well as the experiments reported there (Hearst 1997). The texts were not the same but chosen from the same genre (expository text from a popular science magazine).

More recently, the Topic Detection and Tracking (TDT) Project has been developed as an evaluation standard for related tasks (TDT 1997a, 1997b). The tasks are not the same, however. Here, it is the detection of topic shifts within documents, whereas the TDT Project aims at separating and classifying documents in a continuous stream. It will still be interesting to compare these methods with others in the TDT context, but that will be a different project.

Finally, the evaluation of results in this task is complicated by the fact that "near hits" (cases in which a section break is off by just one paragraph) do not have any positive effect on the score. This problem has been dealt with in the TDT Project by a more flexible score that becomes gradually worse as the distance between hypothesized and real boundaries increases. A reevaluation and comparison of various methods using this kind of measure is left for another occasion.

ACKNOWLEDGMENTS

I am grateful to Stanley Peters, Yasuhiro Takayama, Hinrich Schütze, David Beaver, Edward Flemming, and three anonymous reviewers for helpful discussion and comments, to Stanley Peters for office space and computational infrastructure, and to Raymond Flournoy for assistance with the context vectors. Most of this material was presented at the 1999 meeting of the Pacific Association for Computational Linguistics (PACLING) in Waterloo, Ontario (Kaufmann 1999).

REFERENCES

BENBRAHIM, M., and K. AHMAD. 1995. Text summarisation: The role of lexical cohesion analysis. The New Review of Document and Text Management, **1**:321–335.

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

DEERWESTER, S., S. T. DUMAIS, G. W. FURNAS, and T. K. LANDAUER. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science, **41**(6):391–407.

DENT, M., and R. E. MERCER. 1999. A comparison of word relatedness measures. *In* Proceedings of PACLING'99. *Edited by* Nick Cercone, Kiyoshi Kogure, and Kanlaya Naruedomkul, pp. 270–275.

GOLUB, G. H., and C. F. VAN LOAN. 1989. Matrix computations. Johns Hopkins University Press, Baltimore.

GREEN, S. J. 1997. Automatically generating hypertext by computing semantic similarity. Ph.D. thesis, University of Toronto, Canada.

HAHN, U. 1990. Topic parsing: Accounting for text macro structures in full-text analsysis. Information Processing and Management, **26**:135–170.

HALLIDAY, M. A., and R. HASAN. 1976. Cohesion in English. Longman, London.

HEARST, M. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics, **23**(1):33–64.

HEINONEN, O. 1999. An experimental comparison of several quantitative evaluation schemes for text segmentation. Manuscript, University of Helsinki, Finland.

HOEY, M. 1991. Patterns of Lexis in Text. Oxford University Press, New York.

KAUFMANN, S. 1999. Second-order cohesion: Using context vectors in text segmentation. *In* Proceedings of PACLING'99. University of Waterloo, Canada, pp. 209–222.

KOZIMA, H. 1993. Computing lexical cohesion as a tool for text analysis. Dissertation, University of Electro-Communications, Tokyo. Available online at `http://www-karc.crl.go.jp/kss/xkozima/work/` [1998, November 30].

LEE, L. 1999. Measures of distributional similarity. *In* Proceedings of ACL'37. College Park, MD, pp. 25–32.

LITMAN, D. J., and R. J. PASSONNEAU. 1995. Combining multiple knowledge sources for discourse segmentation. *In* Proceedings of ACL'37. College Park, MD, pp. 108–115.

MORRIS, J., and G. HIRST. 1991. Lexical cohesion computed by thesaural relations as an indication of the structure of text. Computational Linguistics, **17**(1):21–48.

REYNAR, J. C. 1998. Topic segmentation: Algorithms and applications. Dissertation, University of Pennsylvania, Philadelphia. Available online at `http://www.cis.edu/~jcreynar/research.html` [1999, April 24].

SALTON, G., and J. ALLEN. 1993. Selective text utilization and text traversal. *In* Proceedings of the Hypertext '93, Association for Computing Machinery, Seattle, 131–144.

SALTON, G., A. SINGHAL, C. BUCKLEY, and M. MITRA. 1996. Automatic text decomposition using text segments and text themes. *In* Proceedings of the Hypertext '98, Association for Computing Machinery, Washington, D.C., pp. 53–65.

SCHÜTZE, H. 1997. Ambiguity Resolution in Language Learning, Stanford, CA. CSLI.

SCHÜTZE, H. 1998. Automatic word sense discrimination. Computational Linguistics, **24**(1):97–123.

TDT. 1997a. The TDT pilot study corpus documentation version 1.3. Distributed by the Linguistic Data Consortium, University of Pennsylvania, Philadelphia.

TDT. 1997b. The topic detection and tracking (TDT) pilot study evaluation plan. Distributed by the Linguistic Data Consortium, University of Pennsylvania, Philadelphia.

YOUMANS, G. 1991. A new tool for discourse analysis: The vocabulary-management profile. Language, **47**(4):763–789.