Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning

Vasileios Hatzivassiloglou[*], **Judith L. Klavans**[*†], and **Eleazar Eskin**[*]

[*]Department of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, N.Y. 10027

[†]Center for Research on Information Access
Columbia University
535 West 114th Street
New York, N.Y. 10027

{vh, klavans, eeskin}@cs.columbia.edu

Abstract

We present a new composite similarity metric that combines information from multiple linguistic indicators to measure semantic distance between pairs of small textual units. Several potential features are investigated and an optimal combination is selected via machine learning. We discuss a more restrictive definition of similarity than traditional, document-level and information retrieval-oriented, notions of similarity, and motivate it by showing its relevance to the multi-document text summarization problem. Results from our system are evaluated against standard information retrieval techniques, establishing that the new method is more effective in identifying closely related textual units.

1 Research Goals

In this paper, we focus on the problem of detecting whether two small textual units (paragraph- or sentence-sized) contain common information, as a necessary step towards extracting such common information and constructing thematic groups of text units across multiple documents. Identifying similar pieces of text has many applications (e.g., summarization, information retrieval, text clustering). Most research in this area has centered on detecting similarity between documents [Willet 1988], similarity between a query and a document [Salton 1989] or between a query and a segment of a document [Callan 1994]. While effective techniques have been developed for document clustering and classification which depend on inter-document similarity measures, these techniques mostly rely on shared words, or occasionally collocations of words [Smeaton 1992]. When larger units of text are compared, overlap may be sufficient to detect similarity; but when the units of text are small, simple surface matching of words and phrases is less likely to succeed since the number of potential matches is smaller.

Our task differs from typical text matching applications not only in the smaller size of the text units compared, but also in its overall goal. Our notion of similarity is more restrictive than topical similarity—we provide a detailed definition in the next section. We aim to recover sets of small textual units from a collection of documents so that each text unit within a given set describes the same action. Our system, which is fully implemented, is further motivated by the need for determining similarity between small pieces of text across documents that potentially span different topics during multi-document summarization. It serves as the first component of a domain-independent multi-document summarization system [McKeown *et al.* 1999] which generates a summary through text reformulation [Barzilay *et al.* 1999] by combining information from these similar text passages.

We address concerns of sparse data and the narrower than topical definition of similarity by exploring several linguistic features, in addition to shared words or collocations, as indicators of text similarity. Our *primitive* features include linked noun phrases, WordNet synonyms, and semantically similar verbs. We also define *composite* features over pairs of primitive features. We then provide an effective method for aggregating the feature values into a similarity measure using machine learning, and present results

on a manually annotated corpus of 10,345 pairs of compared paragraphs. Our new features, and especially the composite ones, are shown to outperform traditional techniques such as TF*IDF [Buckley 1985; Salton 1989] for determining similarity over small text units.

2 Definition of Similarity

Similarity is a complex concept which has been widely discussed in the linguistic, philosophical, and information theory communities. For example, Frawley [1992] discusses all semantic typing in terms of two mechanisms: the detection of similarity and difference. Jackendoff [1983] argues that standard semantic relations such as synonymy, paraphrase, redundancy, and entailment all result from judgments of likeness whereas antonymy, contradiction, and inconsistency derive from judgments of difference. Losee [1998] reviews notions of similarity and their impact on information retrieval techniques.

For our task, we define two text units as similar if they share the same focus on a common concept, actor, object, or action. In addition, the common actor or object must perform or be subjected to the same action, or be the subject of the same description. For example, Figure 1 shows three input text fragments (paragraphs) taken from the TDT pilot corpus (see Section 5.1), all from the same topic on the forced landing of a U.S. helicopter in North Korea.

We consider units (a) and (b) in Figure 1 to be similar, because they both focus on the same event (loss of contact) with the same primary participant (the helicopter). On the other hand, unit (c) in Figure 1 is not similar to either (a) or (b). Although all three refer to a helicopter, the primary focus in (c) is on the emergency landing rather than the loss of contact.

We discuss an experimental validation of our similarity definition in Section 5.2, after we introduce the corpus we use in our experiments.

3 Related Work

Although there is related empirical research on determining text similarity, primarily in the information retrieval community, there are two major differences between the goals of this earlier work and the problem we address in this paper. First, the notion of similarity as defined in the previous section is more restrictive than the traditional definition of similarity [Anderberg 1973; Willet 1988]. Standard notions of similarity generally involve the creation of a vector or profile of characteristics of a text fragment, and then computing on the basis of frequencies the distance between vectors to determine conceptual distance [Salton and Buckley 1988; Salton 1989]. Features typically include stemmed words although sometimes multi-word units and collocations have been used [Smeaton 1992], as well as typological characteristics, such as thesaural features. The distance between vectors for one text (usually a query) and another (usually a document) then determines closeness or similarity [van Rijsbergen 1979]. In some cases, the texts are represented as vectors of sparse n-grams of word occurrences and learning is applied over those vectors [Schapire and Singer 1999]. But since our definition of similarity is oriented to the small-segment goal, we make more fine-grained distinctions. Thus, a set of passages that would probably go into the same class by standard IR criteria would be further separated by our methods.

Second, we have developed a method that functions over pairs of small units of text, so the size of the input text to be compared is different. This differs from document-to-document

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

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

(c) An OH-58 U.S. military scout helicopter made an emergency landing in North Korea at about 9.15 p.m. EST Friday (0215 GMT Saturday), the Defense Department said.

Figure 1: Input text units (from the TDT pilot corpus, topic 11).

or query-to-document comparison. A closely related problem is that of matching a query to the relevant segment from a longer document [Callan 1994; Kaszkiel and Zobel 1998], which primarily involves determining which segment of a longer document is relevant to a query, whereas our focus is on which segments are similar to each other. In both cases, we have less data to compare, and thus have to explore additional or more informative indicators of similarity.

4 Methodology

We compute a feature vector over a pair of textual units, where features are either *primitive*, consisting of one characteristic, or *composite*, consisting of pairs of primitive features.

4.1 Primitive Features

Our features draw on a number of linguistic approaches to text analysis, and are based on both single words and simplex noun phrases (head nouns preceded by optional premodifiers but with no embedded recursion). Each of these morphological, syntactic, and semantic features has several variations. We thus consider the following potential matches between text units:

- **Word co-occurrence**, i.e., sharing of a single word between text units. Variations of this feature restrict matching to cases where the parts of speech of the words also match, or relax it to cases where just the stems of the two words are identical.
- **Matching noun phrases**. We use the LinkIt tool [Wacholder 1998] to identify simplex noun phrases and match those that share the same head.
- **WordNet synonyms**. WordNet [Miller *et al.* 1990] provides sense information, placing words in sets of synonyms (*synsets*). We match words that appear in the same synset. Variations on this feature restrict the words considered to a specific part-of-speech class.
- **Common semantic classes for verbs**. Levin's [1993] semantic classes for verbs have been found to be useful for determining document type and text similarity [Klavans and Kan 1998]. We match two verbs that share the same semantic class.
- **Shared proper nouns**. Proper nouns are identified using the ALEMBIC tool set [Aberdeen *et al.* 1995]. Variations on proper noun matching include restricting the proper noun type to a person, place, or an organization (these subcategories are also extracted with ALEMBIC's named entity finder).

In order to normalize for text length and frequency effects, we experimented with two types of optional normalization of feature values. The first is for text length (measured in words), where each feature value is normalized by the size of the textual units in the pair. Thus, for a pair of textual units A and B, the feature values are divided by:

$$\sqrt{\text{length}(A) \times \text{length}(B)} \qquad (1)$$

This operation removes potential bias in favor of longer text units.

The second type of normalization we examined was based on the relative frequency of occurrence of each primitive. This is motivated by the fact that infrequently matching primitive elements are likely to have a higher impact on similarity than primitives which match more frequently. We perform this normalization in a manner similar to the IDF part of TF*IDF [Salton 1989]. Every primitive element is associated with a value which is the number of textual units in which the primitive appeared in the corpus. For a primitive element which compares single words, this is the number of textual units which contain that word in the corpus; for a noun phrase, this is the number of textual units that contain noun phrases that share the same head; and similarly for other primitive types. We multiply each feature's value by:

$$\log \frac{\text{Total number of textual units}}{\text{Number of textual units containing this primitive}} \qquad (2)$$

Since each normalization is optional, there are four variations for each primitive feature.

4.2 Composite Features

In addition to the above *primitive* features that compare single items from each text unit, we use *composite* features which combine pairs of primitive features. Composite features are defined by placing different types of restrictions on the participating primitive features:

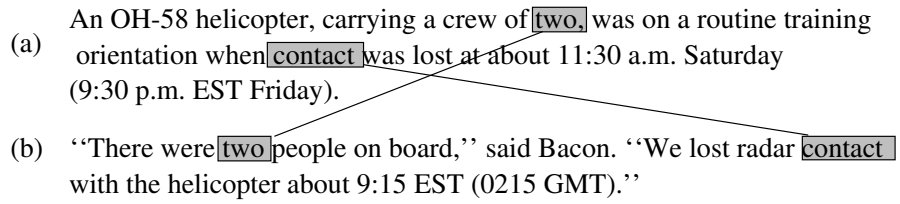

Figure 2: A composite feature over word primitives with a restriction on order would count the pair "two" and "contact" as a match because they occur with the same relative order in both textual units.

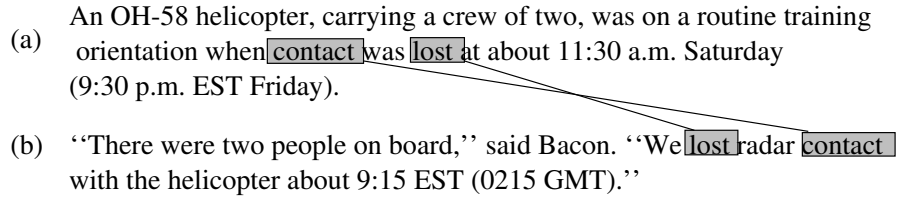

Figure 3: A composite feature over word primitives with a restriction on distance would match on the pair "lost" and "contact" because they occur within two words of each other in both textual units.

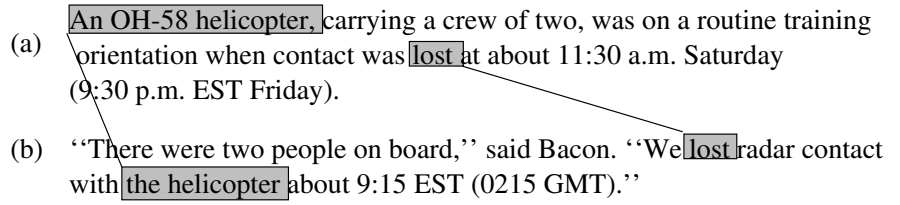

Figure 4: A composite feature with restrictions on the primitives' type. One primitive must be a matching simplex noun phrase (in this case, a helicopter), while the other primitive must be a matching verb (in this case, "lost".) The example shows a pair of textual units where this composite feature detects a valid match.

- **Ordering**. Two pairs of primitive elements are required to have the same relative order in both textual units (see Figure 2).
- **Distance**. Two pairs of primitive elements are required to occur within a certain distance in both textual units (see Figure 3). The maximum distance between the primitive elements can vary as an additional parameter. A distance of one matches rigid collocations whereas a distance of five captures related primitives within a region of the text unit [Smeaton 1992; Smadja 1993].
- **Primitive**. Each element of the pair of primitive elements can be restricted to a specific primitive, allowing more expressiveness in the composite features. For example, we can restrict one of the primitive features to be a simplex noun phrase and the other to be a verb; then, two noun phrases, one from each text unit, must match according to the rule for matching simplex noun phrases (i.e., sharing the same head), and two verbs must match according to the rule for verbs (i.e., sharing the same semantic class); see Figure 4.[1] This particular combination loosely approximates grammatical relations, e.g., matching subject-verb pairs.

[1] Verbs can also be matched by the first (and more restrictive) rule of Section 4.1, namely requiring that their stemmed forms be identical.

Since these restrictions can be combined, many different composite features can be defined, although our empirical results indicate that the most successful tend to include a distance constraint. As we put more restrictions on a composite feature, the fewer times it occurs in the corpus; however, some of the more restrictive features are most effective in determining similarity. Hence, there is a balance between the discriminatory power of these features and their applicability to a large number of cases. Composite features are normalized as primitive features are (i.e., for text unit length and for frequency of occurrence). This type of normalization also uses equation (2) but averages the normalization values of each primitive in the composite feature.

4.3 Learning a Classifier

For each pair of text units, we compute a vector of primitive and composite feature values. To determine whether the units match overall, we employ a machine learning algorithm, RIPPER [Cohen 1996], a widely used and effective rule induction system. RIPPER is trained over a corpus of manually marked pairs of units; we discuss the specifics of our corpus and of the annotation process in the next session. We experiment with varying RIPPER's *loss ratio*, which measures the cost of a false positive relative to that of a false negative (where we view "similar" as the positive class), and thus controls the relative weight of precision versus recall. This is an important step in dealing with the sparse data problem; most text units are not similar, given our restrictive definition, and thus positive instances are rare.

5 Results

5.1 The Evaluation Corpus

For evaluation, we use a set of articles already classified into topical subsets which we obtained from the Reuters part of the 1997 pilot Topic Detection and Tracking (TDT) corpus. The TDT corpus, developed by NIST and DARPA, is a collection of 16,000 news articles from Reuters and CNN where many of the articles and transcripts have been manually grouped into 25 categories each of which corresponds to a single event (see `http://morph.ldc.upenn.edu/Catalog/LDC98T25.html`). Using the Reuters part of the corpus, we selected five of the larger categories and extracted all articles assigned to them from several randomly chosen days, for a total of 30 articles.

Since paragraphs in news stories tend to be short—typically one or two sentences—in this study we use paragraphs as our small text units, although sentences would also be a possibility. In total, we have 264 text units and 10,345 comparisons between units. As comparisons are made between all pairs of paragraphs from the same topic, the total number of comparisons is equal to

$$\sum_{i=1}^{5} \binom{N_i}{2}$$

where N_i is the number of paragraphs in all selected articles from topical category i.

Training of our machine learning component was done by three-fold cross-validation, randomly splitting the 10,345 pairs of paragraphs into three (almost) equally-sized subsets. In each of the three runs, two of these subsets were used for training and one for testing.

To create a reference standard, the entire collection of 10,345 paragraph pairs was marked for similarity by two reviewers who were given our definition and detailed instructions. Each reviewer independently marked each pair of paragraphs as similar or not similar. Subsequently, the two reviewers jointly examined cases where there was disagreement, discussed reasons, and reconciled the differences.

5.2 Experimental Validation of the Similarity Definition

In order to independently validate our definition of similarity, we performed two additional experiments. In the first, we asked three additional judges to determine similarity for a random sample of 40 paragraph pairs. High agreement between judges would indicate that our definition of similarity reflects an objective reality and can be mapped unambiguously to an operational procedure for marking text units as similar or not. At the same time, it would also validate the judgments between text units that we use for our experiments (see Section 5.1). In this task, judges were given the opportunity to provide reasons for claiming similarity or dissimilarity, and comments on the task were logged for future analysis. The three additional

judges agreed with the manually marked and standardized corpus on 97.6% of the comparisons.

Unfortunately, approximately 97% (depending on the specific experiment) of the comparisons in both our model and the subsequent validation experiment receive the value "not similar". This large percentage is due to our fine-grained notion of similarity, and is parallel to what happens in randomly sampled IR collections, since in that case most documents will not be relevant to any given query. Nevertheless, we can account for the high probability of inter-reviewer agreement expected by chance, $0.97 \cdot 0.97 + (1-0.97) \cdot (1-0.97) = 0.9418$, by referring to the kappa statistic [Cohen 1960; Carletta 1996]. The kappa statistic is defined as

$$K = \frac{P_A - P_0}{1 - P_0}$$

where P_A is the probability that two reviewers agree in practice, and P_0 is the probability that they would agree solely by chance. In our case, $P_A = 0.976$, $P_0 = 0.9418$, and $K = 0.5876$, indicating that the observed agreement by the reviewers is indeed significant.[2] If P_0 is estimated from the particular sample used in this experiment rather than from our entire corpus, it would be only 0.9, producing a value of 0.76 for K.

In addition to this validation experiment that used randomly sampled pairs of paragraphs (and reflected the disproportionate rate of occurrence of dissimilar pairs), we performed a balanced experiment by randomly selecting 50 of the dissimilar pairs and 50 of the similar pairs, in a manner that guaranteed generation of an independent sample.[3] Pairs in this subset were rated for similarity by two additional independent reviewers, who agreed on their decisions 91% of the time, versus 50% expected by chance; in this case, $K = 0.82$. Thus, we feel confident in the reliability of our annotation process, and can use the annotated corpus to assess the performance of our similarity measure and compare it to measures proposed earlier in the information retrieval literature.

[2] K is always between 0 and 1, with 0 indicating no better agreement than expected by chance and 1 indicating perfect agreement.

[3] To guarantee independence, pairs of paragraphs were randomly selected for inclusion in the sample but a pair (A, B) was immediately rejected if there were paragraphs $X_1, \ldots, X_n$ for $n \geq 0$ such that all pairs $(A, X_1), (X_1, X_2), \ldots, (X_n, B)$ had already been included in the sample.

5.3 Performance Comparisons

We compare the performance of our system to three other methods. First, we use standard TF*IDF, a method that with various alterations, remains at the core of many information retrieval and text matching systems [Salton and Buckley 1988; Salton 1989]. We compute the total frequency (TF) of words in each text unit. We also compute the number of units each word appears in in our training set (DF, or document frequency). Then each text unit is represented as a vector of TF*IDF scores calculated as

$$TF(\text{word}_i) \cdot \log \frac{\text{Total number of units}}{DF(\text{word}_i)}$$

Similarity between text units is measured by the cosine of the angle between the corresponding two vectors (i.e., the normalized inner product of the two vectors). A further cutoff point is selected to convert similarities to hard decisions of "similar" or "not similar"; different cutoffs result in different tradeoffs between recall and precision.

Second, we compare our method against a standard, widely available information retrieval system developed at Cornell University, SMART [Buckley 1985].[4] SMART utilizes a modified TF*IDF measure (ATC) plus stemming and a fairly sizable stopword list.

Third, we use as a baseline method the default selection of the most frequent category, i.e., "not similar". While this last method cannot be effectively used to identify similar paragraphs, it offers a baseline for the overall accuracy of any more sophisticated technique for this task.

5.4 Experimental Results

Our system was able to recover 36.6% of the similar paragraphs with 60.5% precision, as shown in Table 1. In comparison, the unmodified TF*IDF approach obtained only 32.6% precision when recall is 39.1%, i.e., close to our system's recall; and only 20.8% recall at precision of 62.2%, comparable to our classifier's

[4] We used version 11.0 of SMART, released in July 1992.

	Recall	Precision	Accuracy
Machine learning over linguistic indicators	36.6%	60.5%	98.8%
TF*IDF	30.0%	47.4%	97.2%
SMART	29.1%	48.3%	97.1%
Default choice (baseline)	0%	undefined	97.5%

Table 1: Experimental results for different similarity metrics. For comparison purposes, we list the average recall, precision, and accuracy obtained by TF*IDF and SMART at the two points in the precision-recall curve identified for each method in the text (i.e., the point where the method's precision is most similar to ours, and the point where its recall is most similar to ours).

precision. SMART (in its default configuration) offered only a small improvement over the base TF*IDF implementation, and significantly underperformed our method, obtaining 34.1% precision at recall of 36.7%, and 21.5% recall at 62.4% precision. The default method of always marking a pair as dissimilar obtains of course 0% recall and undefined precision. Figure 5 illustrates the difference between our system and straight TF*IDF at different points of the precision-recall spectrum.

When overall accuracy (total percentage of correct answers over both categories of similar and non-similar pairs) is considered, the numbers are much closer together: 98.8% for our approach; 96.6% and 97.8% for TF*IDF on the two P-R points mentioned for that method above; 96.5% and 97.6% for SMART, again at the two P-R points mentioned for SMART earlier; and 97.5% for the default baseline.[5] Nevertheless, since the challenge of identifying sparsely occurring similar small text units is our goal, the accuracy measure and the baseline technique of classifying everything as not similar are included only for reference but do not reflect our task.

[5]Statistical tests of significance cannot be performed for comparing these values, since paragraphs appear in multiple comparisons and consequently the comparisons are not independent.

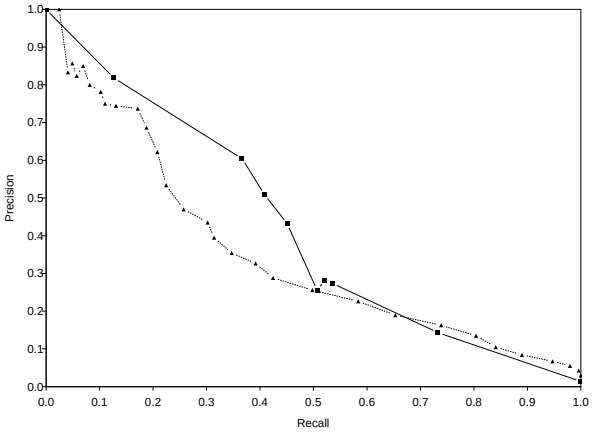

Figure 5: Precision-recall graph comparing our method using RIPPER (solid line with squares) versus TF*IDF (dotted line with triangles).

6 Analysis and Discussion of Feature Performance

We computed statistics on how much each feature helps in identifying similarity, summarized in Table 2. Primitive features are named according to the type of the feature (e.g., *Verb* for the feature that counts the number of matching verbs according to exact matches). Composite feature names indicate the restrictions applied to primitives. For example, the composite feature *Distance* $\leq$ *4* restricts a pair of matching primitives to occur within a relative distance of four words. If the composite feature also restricts the types of the primitives in the pair, the name of the restricting primitive feature is added to the composite feature name. For example the feature named *Verb Distance* $\leq$ *5* requires one member of the pair to be a verb and the relative distance between the primitives to be at most five.

The second column in Table 2 shows whether the feature value has been normalized according to its overall rarity[6], while the third column indicates the actual threshold used in decisions assuming that only this feature is used for classification. The fourth column shows the *applicability* of that feature, that is, the percentage of

[6]All results reported in Table 2 include our first normalization step that accounts for the difference in the length of text units.

Feature Name	Normalized?	Threshold	Applicability	Recall	Precision
Any word	Yes	0.360	2.2%	31.4%	41.8%
		0.505	0.6%	16.7%	75.4%
Noun	Yes	0.150	8.1%	43.2%	15.9%
		0.275	1.5%	20.9%	37.0%
Proper noun	Yes	0.200	0.2%	2.0%	30.8%
Verb	No	0.775	1.6%	10.6%	19.7%
Simplex NP	Yes	0.150	5.7%	35.5%	18.6%
		0.275	2.7%	10.1%	44.6%
		0.350	0.7%	3.7%	69.2%
Semantic class of verbs	No	0.875	0.1%	2.0%	3.4%
WordNet	Yes	0.250	5.4%	4.1%	2.3%
Distance ≤ 2	Yes	0.075	4.7%	24.9%	15.7%
Distance ≤ 3	Yes	0.250	0.5%	10.2%	55.6%
Distance ≤ 4	Yes	0.275	1.9%	14.6%	50.0%
Distance ≤ 5	Yes	0.200	1.9%	22.4%	53.4%
Order Distance ≤ 5	Yes	0.200	1.5%	20.4%	40.7%
Noun Distance ≤ 5	Yes	0.175	1.9%	21.2%	31.9%
Verb Distance ≤ 5	Yes	0.200	0.3%	7.3%	66.7%
	No	0.850	0.6%	11.0%	56.3%

Table 2: Statistics for a selected subset of features. Performance measures are occasionally given multiple times for the same feature and normalization option, highlighting the effect of different decision thresholds.

paragraph pairs for which this feature would apply (i.e., have a value over the specified threshold). Finally, the fifth and sixth columns show the recall and precision on identifying similar paragraphs for each independent feature. Note that some features have low applicability over the entire corpus, but target the hard-to-find similar pairs, resulting in significant gains in recall and precision.

Table 2 presents a selected subset of primitive and composite features in order to demonstrate our results. For example, it was not surprising to observe that the most effective primitive features in determining similarity are *Any word*, *Simplex NP*, and *Noun* while other primitives such as *Verb* were not as effective independently. This is to be expected since nouns name objects, entities, and concepts, and frequently exhibit more sense constancy. In contrast, verbs are functions and tend to shift senses in a more fluid fashion depending on context. Furthermore, our technique does not label phrasal verbs (e.g. look up, look out, look over, look for, etc.), which are a major source of verbal ambiguity in English.

Whereas primitive features viewed independently might not have a directly visible effect on identifying similarity, when used in composite features they lead to some novel results. The most pronounced case of this is for *Verb*, which, in the composite feature *Verb Distance* $\leq$ *5*, can help identify similarity effectively, as seen in Table 2. This composite feature approximates verb-argument and verb-collocation relations, which are strong indicators of similarity. At the same time, the more restrictive a feature is, the fewer occurrences of that feature appear in the training set. This suggests that we could consider adding additional features suggested by current results in order to further refine and improve our similarity identification algorithm.

7 Conclusion and Future Work

We have presented a new method to detect similarity between small textual units, which combines primitive and composite features using machine learning. We validated our similarity definition using human judges, applied

our method to a substantial number of paragraph pairs from news articles, and compared results to baseline and standard information retrieval techniques. Our results indicate that our method outperforms the standard techniques for detecting similarity, and the system has been successfully integrated into a larger multiple-document summarization system [McKeown *et al.* 1999].

We are currently working on incorporating a clustering algorithm in order to give as output a set of textual units which are mutually similar rather than just pairwise similar. Future work includes testing on textual units of different size, comparing with additional techniques proposed for document similarity in the information retrieval and computational linguistics literature, and extending the feature set to incorporate other types of linguistic information in the statistical learning method.

Acknowledgments

We are grateful to Regina Barzilay, Hongyan Jing, Kathy McKeown, Shimei Pan, and Yoram Singer for numerous discussions of earlier versions of this paper and for their help with setting up and running RIPPER and SMART. This research has been supported in part by an NSF STIMULATE grant, IRI-96-1879. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[Aberdeen *et al.* 1995] John Aberdeen, John Burger, David Day, Lynette Hirschman, Patricia Robinson, and Marc Vilain. MITRE: Description of the Alembic System as Used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995.

[Anderberg 1973] Michael R. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, 1973. Revised version of the author's thesis, University of Texas at Austin, 1971.

[Barzilay *et al.* 1999] Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. Information Fusion in the Context of Multi-Document Summarization. In *Proceedings of the 37th Annual Meeting of the ACL*, College Park, Maryland, June 1999 (to appear). Association for Computational Linguistics.

[Buckley 1985] Christopher Buckley. Implementation of the SMART Information Retreival System. Technical Report 85-686, Cornell University, 1985.

[Callan 1994] Jaime P. Callan. Passage-Level Evidence in Document Retrieval. In *Proceedings of the 17th ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 302–309, Dublin, Ireland, 1994.

[Carletta 1996] Jean Carletta. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, **22**(2):249–254, June 1996.

[Cohen 1960] Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, **20**:37–46, 1960.

[Cohen 1996] William Cohen. Learning Trees and Rules with Set-Valued Features. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96)*. American Association for Artificial Intelligence, 1996.

[Frawley 1992] William Frawley. *Linguistic Semantics*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1992.

[Jackendoff 1983] Ray Jackendoff. *Semantics and Cognition*. MIT Press, Cambridge, Massachusetts, 1983.

[Kaszkiel and Zobel 1998] Marcin Kaszkiel and Justin Zobel. Passage Retrieval Revisited. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, August 1998.

[Klavans and Kan 1998] Judith L. Klavans and Min-Yen Kan. The Role of Verbs in Document Access. In *Proceedings of the*

36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL/COLING-98), Montreal, Canada, 1998.

[Levin 1993] Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, Illinois, 1993.

[Losee 1998] Robert M. Losee. *Text Retrieval and Filtering: Analytic Models of Performance*. Kluwer Academic Publishers, Boston, Massachusetts, 1998.

[McKeown *et al.* 1999] Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-99)*, Orlando, Florida, 1999 (to appear). American Association for Artificial Intelligence.

[Miller *et al.* 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to WordNet: An On-Line Lexical Database. *International Journal of Lexicography*, **3**(4):235–312, 1990.

[Salton and Buckley 1988] Gerard Salton and Christopher Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, **25**(5):513–523, 1988.

[Salton 1989] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.

[Schapire and Singer 1999] Robert E. Schapire and Yoram Singer. BoosTexter: A Boosting-Based System for Text Categorization. *Machine Learning*, 1999 (to appear).

[Smadja 1993] Frank Smadja. Retrieving Collocations from Text: Xtract. *Computational Linguistics*, **19**(1):143–177, March 1993.

[Smeaton 1992] Alan F. Smeaton. Progress in the Application of Natural Language Processing to Information Retrieval Tasks. *The Computer Journal*, **35**(3):268–278, 1992.

[van Rijsbergen 1979] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[Wacholder 1998] Nina Wacholder. Simplex NPs Clustered by Head: A Method For Identifying Significant Topics in a Document. In *Proceedings of the Workshop on the Computational Treatment of Nominals*, pages 70–79, Montreal, Canada, October 1998. COLING-ACL.

[Willet 1988] Peter Willet. Recent Trends in Hierachical Document Clustering. *Information Processing and Management*, **24**(5):577–597, 1988.