Multiple Similarity Measures and Source-Pair Information in Story Link Detection

Francine Chen **Ayman Farahat** **Thorsten Brants**
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304
{fchen, farahat}@parc.com, thorsten@brants.net

Abstract

State-of-the-art story link detection systems, that is, systems that determine whether two stories are about the same event or *linked*, are usually based on the cosine-similarity measured between two stories. This paper presents a method for improving the performance of a link detection system by using a variety of similarity measures and using source-pair specific statistical information. The utility of a number of different similarity measures, including cosine, Hellinger, Tanimoto, and clarity, both alone and in combination, was investigated. We also compared several machine learning techniques for combining the different types of information. The techniques investigated were SVMs, voting, and decision trees, each of which makes use of similarity and statistical information differently. Our experimental results indicate that the combination of similarity measures and source-pair specific statistical information using an SVM provides the largest improvement in estimating whether two stories are linked; the resulting system was the best-performing link detection system at TDT-2002.

1 Introduction

Story link detection, as defined in the Topic Detection and Tracking (TDT) competition sponsored by the DARPA TIDES program, is the task of determining whether two stories, such as news articles and/or radio broadcasts, are about the same event, or *linked*. In TDT an *event* is defined as "something that happens at some specific time and place" (TDT, 2002). For example, a story about a tornado in Kansas in May and another story about a tornado in Nebraska in June should not be classified as linked because they are about different events, although they both fall under the same general "topic" of natural disasters. But a story about damage due to a tornado in Kansas and a story about the clean-up and repairs due to the same tornado in Kansas are considered linked events.

In the TDT link detection task, a link detection system is given a sequence of time-ordered sets of stories, where each set is from one news source. The system can "look ahead" N source files from the current source file being processed when deciding whether the current pair is linked. Because the TDT link detection task is focused on streams of news stories, one of the primary differences between link detection and the more traditional IR categorization task is that *new* events occur relatively frequently and comparisons of interest are focused on events that are not known in advance. One consequence of this is that the best-performing systems usually adapt to new input. Link detection is thought of as the basis for other event-based topic analysis tasks, such as topic tracking, topic detection, and first-story detection (TDT, 2002).

2 Background and Related Work

The DARPA TDT story link detection task requires identifying pairs of linked stories. The original language of the stories are in English, Mandarin and Arabic. The sources include broadcast news and newswire. For the required story link detection task, the research groups tested their systems on a processed version of the data in which the story boundaries have been manually identified, the Arabic and Mandarin stories have been automatically translated to English, and the broadcast news stories have been converted to text by an automatic speech recognition (ASR) system.

A number of research groups have developed story link detection systems. The best current technology for link detection relies on the use of cosine similarity between document terms vectors with TF-IDF term weighting. In a TF-IDF model, the frequency of a term in a document (TF) is weighted by the inverse document frequency

(IDF), the inverse of the number of documents containing a term. UMass (Allan et al., 2000) has examined a number of similarity measures in the link detection task, including weighted sum, language modeling and Kullback-Leibler divergence, and found that the cosine similarity produced the best results. More recently, in Lavrenko et al. (2002), UMass found that the clarity similarity measure performed best for the link detection task. In this paper, we also examine a number of similarity measures, both separately, as in Allan et al. (2000), and in combination. In the machine learning field, classifier combination has been shown to provide accuracy gains (e.g., Belkin et al.(1995); Kittler et al. (1998); Brill and Wu (1998); Dietterich (2000)). Motivated by the performance improvement observed in these studies, we explored the combination of similarity measures for improving Story Link Detection.

CMU hypothesized that the similarity between a pair of stories is influenced by the *source* of each story. For example, sources in a language that is translated to English will consistently use the same terminology, resulting in greater similarity between linked documents with the same native language. In contrast, sources from radio broadcasts may be transcribed much less consistently than text sources due to recognition errors, so that the expected similarity of a radio broadcast and a text source is less than that of two text sources. They found that similarity *thresholds* that were dependent on the type of the story-pair sources (e.g., English/non-English language and broadcast news/newswire) improved story-link detection results by 15% (Carbonell et al., 2001). We also investigate how to make use of differences in similarity that are dependent on the types of sources composing a story pair. We refer to the statistics characterizing story pairs with the same source types as *source-pair specific information*. In contrast to the source-specific thresholds used by CMU, we *normalize the similarity measures* based on the source-pair specific information, simultaneously with combining different similarity measures.

Other researchers have successfully used machine learning algorithms such as support vector machines (SVM) (Cristianini and Shawe-Taylor, 2000; Joachims, 1998) and boosted decision stumps (Schapire and Singer, 2000) for text categorization. SVM-based systems, such as that described in (Joachims, 1998), are typically among the best performers for the categorization task. However, attempts to directly apply SVMs to TDT tasks such as tracking and link detection have not been successful; this has been attributed in part to the lack of enough data for training the SVM[1]. In these systems, the input was the set of term vectors characterizing each document, similar to the input used for the categorization task. In this paper, we present a method for using SVMs to improve link detection performance by combining heterogeneous input features, composed of multiple similarity metrics and statistical characterization of the story sources. We additionally examine the utility of the statistical information by comparing against decision trees, where the statistical characterization is not utilized. We also examine the utility of the similarity values by comparing against voting, where the classification based on each similarity measure is combined.

[1]http://www.ldc.upenn.edu/Projects/TDT3/email/email_402.html, accessed Mar 11, 2004.

3 System Description

To determine whether two documents are linked, state-of-the-art link detection systems perform three primary processing steps:

1. preprocessing to create a normalized set of terms for representing each document as a vector of term counts, or *term vector*
2. adapting model parameters (i.e., IDF) as new story sets are introduced and computing the similarity of the term vectors
3. determining whether a pair of stories are linked based on the similarity score.

In this paper, we describe our investigations in improving the basic story link detection systems by using source specific information and combining a number of similarity measures. As in the basic story link detection system, a similarity score between two stories is computed. In contrast to the basic story link detection system, a variety of similarity measures is computed and the prediction models use source-pair-specific statistics (i.e., median, average, and variance of the story pair similarity scores). We do this in a post-processing step using machine learning classifiers (i.e., SVMs, decision trees, or voting) to produce a decision with an associated confidence score as to whether a pair of stories are linked. Source-pair-specific statistics and multiple similarity measures are used as input features to the machine learning based techniques in post-processing the similarity scores. In the next sections, we describe the components and processing performed by our system.

3.1 Preprocessing

For preprocessing, we tokenize the data, remove stopwords, replace spelled-out numbers by digits, replace the tokens by their stems using the Inxight LinguistX morphological analyzer, and then generate a term-frequency vector to represent each story. For text where the original source is Mandarin, some of the terms are untranslated. In our experiments, we retain these terms because many are content words. Both the training data and test data are preprocessed in the same way.

3.1.1 Stop Words

Our base stoplist is composed of 577 terms. We extend the stoplist with terms that are represented differently by ASR systems and text documents. For example, in the broadcast news documents in the TDT collection "30" is spelled out as "thirty" and "CNN" is represented as three separate tokens "C", "N", and "N". To handle these differences, an "ASR stoplist" was automatically created. Chen et al. (2003) found that the use of an enhanced stoplist, formed from the union of a base stoplist and ASR stoplist, was very effective in improving performance and empirically better than normalizing ASR abbreviations.

3.1.2 Source-specific Incremental TF-IDF Model

The training data is used to compute the initial document frequency over the corpus for each term. The document frequency of term t, $df(t)$ is defined to be:

$$df(t) = \frac{n(t)}{N} = \frac{\#\ of\ docs\ term\ t\ occurs\ in}{\#\ of\ docs\ in\ corpus}$$

Separate document term counts, $n(t)$, and document counts, N, are computed for each type of source.

Our similarity calculations of documents are based on an incremental TF-IDF model. Term vectors are created for each story, and the vectors are weighted by the inverse document frequency, IDF, i.e., $\log \frac{N}{n(t)}$. In the incremental model, $n(t)$ and N are updated with each new set of stories in a source file. When the k^{th} set of test documents, C_k, is added to the model, the document term counts are updated as:

$$n_k(t) = n_{k-1}(t) + n_{C_k}(t)$$

where $n_{C_k(t)}$ denotes the document count for term t in the newly added set of documents C_k. The initial document counts $n_0(t)$ were generated from a training set. In a static TF-IDF model, new words (i.e., those words, that did not occur in the training set) are ignored in further computations. An incremental TF-IDF model uses the new vocabulary in similarity calculations, which is an advantage for the TDT task because new events often contain new vocabulary.

Since very low frequency terms t tend to be uninformative, we set a threshold θ_d such that only terms with $df_{C_k}(t) \geq \theta_d$ are used with sources up through C_k. For these experiments, we used $\theta_d = 2$.

3.1.3 Term Weighting

The document frequencies, $df(t)$, the number of documents containing term t, and document term frequencies, $f(d,t)$, are used to calculate TF-IDF based weights $w(t,d)$ for the terms in a document (or story) d:

$$w(d,t) = \frac{1}{Z(d)} f(d,t) \cdot \log \frac{N}{n(t)} \qquad (1)$$

where N is the total number of documents and $Z(d)$ is a normalization value. For the Hellinger, Tanimoto, and clarity measures, it is computed as:

$$Z(d) = \sum_t f(d,t) \cdot \log \frac{N}{n(t)} \qquad (2)$$

For cosine distance it is computed as:

$$Z(d) = \sqrt{\sum_t \left[f(d,t) \cdot \log \frac{N_t}{n(t)} \right]^2} \qquad (3)$$

3.2 Similarity Measures

In addition to the cosine similarity measure used in baseline systems, we compute story pair similarity over a set of measures, motivated by the accuracy gains obtained by others when combining classifiers (see Section 2). A vector composed of the similarity values is created and is given to a trained classifier, which emits a score. The score can be used as a measure of confidence that the story pairs are linked.

The similarity measures that we examined are cosine, Hellinger, Tanimoto, and clarity. Each of the measures captures a different aspect of the similarity of the terms in a document. Classifier combination has been observed to perform best when the classifiers produce independent judgments. The cosine distance between the word distribution for documents d_1 and d_2 is computed as:

$$sim(d_1, d_2) = \sum_t w(d_1,t) \cdot w(d_2,t)$$

This measure has been found to perform well and was used by all the TDT 2002 link detection systems (unpublished presentations at the TDT2002 workshop).

In contrast to the Euclidean distance based cosine measure, the Hellinger measure is a probabilistic measure. The Hellinger measure between the word distributions for documents d_1 and d_2 is computed as:

$$sim(d_1, d_2) = \sum_t \sqrt{w(d_1,t) \cdot w(d_2,t)}$$

where t ranges over the terms that occur in d_1 or d_2. In Brants et al. (2002), the Hellinger measure was used in a text segmentation application and was found to be superior to the cosine similarity.

The Tanimoto (Duda and Hart, 1973) measure is a measure of the ratio of the number of shared terms between two documents to the number possessed by one document only. We modified it to use frequency counts, instead of a binary indicator as to whether a term is present and computed it as:

$$sim(d_1, d_2) = \frac{\sum_t w(d_1,t) \cdot w(d_2,t)}{\sum_t w(d_1,t)^2 + w(d_2,t)^2 - w(d_1,t) \cdot w(d_2,t)}.$$

The clarity measure was introduced by Croft et al. (2001) and shown to improve link detection performance by Lavrenko et al. (2002). It gets its name from the distance to general English, which is called *Clarity*. We used a symmetric version that is computed as:

$$\begin{aligned} sim(d_1, d_2) \quad = \quad & -KL(w(t,d_1)||w(t,d_2)) \\ & + KL(w(t,d_1)||GE) \\ & - KL(w(t,d_2)||w(t,d_1)) \\ & + KL(w(t,d_2||GE) \end{aligned}$$

where GE is the probability distribution of words for "general English" as derived from the training corpus, and KL is the Kullback-Leibler divergence: $KL(p||q) = \sum_x p(x) log \frac{p(x)}{q(x)}$. In computing the clarity measure, the term frequencies were smoothed with the General English model using a weight of 0.01. This enables the KL divergence to be defined when $w(t,d_1)$ or $w(t,d_2)$ is 0. The idea behind the clarity measure is to give credit to similar pairs of documents with term distributions that are very different from general English, and to discount similar pairs of documents with term distributions that are close to general English, which can be interpreted as being non-topical.

We also defined the "source-pair normalized cosine" distance as the cosine distance normalized by dividing by the running median of the similarity values corresponding to the source-pair:

$$sim(d_1, d_2) = \frac{\sum_t w(d_1, t) * w(d_2, t)}{\text{median}(sim_{d_i, d_j})}$$

where $\text{median}(sim_{d_i,d_j})$ is the running median of the similarity values of all processed story pairs where the source of d_i is the same as d_1 and the source of d_j is the same as d_2. This is a finer-grained use of source pair information than what was used by CMU, which used decision thresholds conditioned on whether or not the sources were cross-language or cross-ASR/newswire (Carbonell et al., 2001).

In a base system employing a single similarity measure, the system computes the similarity measure for each story pair, which is given to the evaluation program (see Section 4.2).

3.3 Improving Story Link Detection Performance

We examined a number of methods for improving link detection, including:

- compare the 5 similarity measures alone
- combine subsets of similarity scores using a support vector machine (SVM)
- combine source-pair statistics with the corresponding similarity score using an SVM, for each of the 5 similarity measures
- combine subsets of similarity scores with source pair information using an SVM
- compare SVMs, decision trees, and majority voting as alternative methods for combining scores

In contrast to earlier attempts that applied the machine learning categorization paradigm of using the term vectors as input features (Joachims, 1998) to the link detection task, we believed that the use of document term vectors is too fine-grained for the SVMs to develop good generalization with a limited amount of labeled training data. Furthermore, the use of terms as input to a learner, as was done in the categorization task (see Section 2), would require frequent retraining of a link detection system since new stories often discuss new topics and introduce new terms. For our work, we used more general characteristics of a document pair, the *similarity* between a pair of documents, as input to the machine learning systems. Thus, in contrast to the term-based systems, the machine learning techniques are used in a *post-processing* step after the similarity scores are computed. Additionally, to normalize differences in expected similarity among pairs of source types, source-pair statistics are used as features in deciding whether two stories are linked and in estimating the confidence of the decision.

In the next sections, we describe our methods for combining the similarity scores using machine learning techniques, and for combining the similarity scores with source-pair specific information.

3.3.1 Combining Similarity Scores with SVMs

We used an SVM to combine sets of similarity measures for predicting whether two stories are linked because theoretically it has good generalization properties (Cristianini and Shawe-Taylor, 2000), it has been shown to be a competitive classifier for a variety of tasks (e.g., (Cristianini and Shawe-Taylor, 2000; Gestal et al., 2000), and it makes full use of the similarity scores and statistical characterizations. We also empirically show in Section 4.3.2 that it provides better performance than decision trees and voting for this task. The SVM is first trained on a set of labeled data where the input features are the sets of similarity measures and the class labels are the manually assigned decisions as to whether a pair of documents are linked. The trained model is then used to automatically decide whether a new pair of stories are linked. For the support vector machine, we used SVM-light (Joachims, 1999). A polynomial kernel was used in all the reported SVM experiments. In addition to making a decision as to whether two stories are linked, we use the value of the decision function produced by SVM-light as a measure of

Table 1: Source Pair Groups

	asr:asr	asr:text	text:text
English:English	a	b	c
English:Arabic	d	e	f
English:Mandarin	g	h	i
Arabic:Arabic	j	k	l
Arabic:Mandarin	m	n	o
Mandarin:Mandarin	p	q	r

confidence, which serves as input to the evaluation program.

Training SVM-light on a 20,000 story-pair training corpus usually requires less than five minutes on a 1.8 GHz Linux machine, although the time is quite variable depending on the corpus characteristics. However, once the system is trained, testing new story pair similarities requires less than 1 min for over 20,000 story pairs.

3.3.2 Source-Pair Specific Information

Source-pair-specific information that statistically characterizes each of the similarity measures is used in a post-processing step. In particular, we compute statistics from the training data similarity scores for different combinations of *source modalities* and *languages*. The *modality* pairs that we considered are: asr:asr, asr:text, and text:text, where asr represents "automatic speech recognition". The combinations of *languages* that we used are: English:English, English:Arabic, English:Mandarin, Arabic:Arabic, Arabic:Mandarin, Mandarin:Mandarin.

The rows of Table 1 represent possible combinations of source language for the story pairs; the columns represent different combinations of source modality. The alphabetic characters in the cells represent the pair similarity statistics of mean, median, and variance for that condition obtained from the training corpus. For conditions where training data was not available, we used the statistics of a coarser grouping. For example, if there is no data for the cell with languages Mandarin:Arabic and modality pair asr:asr, we would use statistics from the language pair non-English:non-English and modality pair asr:asr.

Prior to use in link detection, an SVM is trained on a set of features computed for each story pair. These include the similarity measures described in Section 3.2 and corresponding source-pair specific statistics (average, median and variance) for the similarity measures. The motivation for using the statistical values is to inform the SVM about the type of source pairs that are being considered. Rather than using categorical labels, the source-pair statistics provide a natural ordering to the source-pair types and can be used for normalization. When a new pair of stories is post-processed, the computed similarity measures and the corresponding source-pair statistics are used as input to the trained SVM.

3.3.3 Other Methods for Combining Similarity Scores

In addition to SVMs, we investigated the utility of decision trees (Breiman et al., 1984) and majority voting (Kittler et al., 1998) as techniques to combine similarity measures and statistical information in a post-processing step. The simplest method that we examined for combining similarity scores is to create a separate classifier for each similarity measure and then classify based a combination of the votes of the different classifiers (Kittler et al., 1998). This method does not utilize statistical information. The single measure classifiers use an empirically determined threshold based on training data.

Decision trees and SVMs are classifiers that use the similarity scores directly. Decision trees such as C4.5 easily handle categorical data. In our experiments, we noted that although source-pair specific statistics were used as an input feature to the decision tree, the decision trees treated the source-pair based statistical information as categorical features. For the decision trees we used the WEKA implementation of C4.5 (Witten and Frank, 1999).

4 Experiments

We conducted a set of experiments to compare the utility of combining similarity measures and the use of normalization statistics. We also compared the utility of different statistical learners.

4.1 Corpora

For our studies, we used corpora developed by the LDC for the TDT tasks (Cieri et al, 2003). The TDT3 corpus contains 40,000 news articles and broadcast news stories with 120 labeled events in English, Mandarin and Arabic from October through December 1998. For our comparative evaluations, we initialized the document term counts and document counts using the TDT2 data (from TDT 1998). Our "post-processing" system was trained on the *TDT3pub* partition of the TDT2001 story pairs, and tested on the *TDT3unp* partition of the TDT2001 test story pairs. The source-pair statistics were computed from the linked story pairs in the TDT2002 dry run test set. There are 20,966, 27,541, and 20,191 labeled story pairs in TDT3pub, TDT2002 dry run, and TDT3unp, respectively. The preprocessing and similarity computations do not require training, although adaptation is performed by incrementally updating the document counts, document frequencies, and the source-specific similarities, and using the updated values in the computations. The training data is used to compute similarity data for training the post-processing systems.

Table 2: Topic-weighted Min Detection Cost for Different Systems

system	min DET
A	0.2368
B	0.3439
C	0.3175
D	0.2606
E	*0.2342*

4.2 Evaluation Measures

The goal of link detection is to minimize the cost, or penalty, due to errors by the system. The TDT tasks are evaluated by computing a "detection cost":

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{non-target}$$

where C_{Miss} is the cost of a miss, P_{Miss} is the estimated probability of a miss, P_{target} is the prior probability that a pair of stories are linked, C_{FA} is the cost of a false alarm, P_{FA} is the estimated probability of a false alarm, and $P_{non-target}$ is the prior probability that a pair of stories are not linked. A miss occurs when a linked story pair is not identified as linked by the system. A false alarm occurs when a pair of stories that are not linked are identified as linked by the system. A target is a pair of linked stories; conversely, a non-target is a pair of stories that are not linked. For the link detection task these parameters are set as follows: C_{Miss} is 1.0, P_{target} is 0.02, and C_{FA} is 0.1. The cost for each topic is equally weighted (i.e., the cost is topic-weighted, rather than story-weighted) and normalized so that for a given system, "$(C_{Det})_{Norm}$ can be no less than one without extracting information from the source data" (TDT, 2002):

$$(C_{Det})_{Norm} = \frac{C_{Det}}{\min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{non-target})}$$

and $C_{Detoverall} = \sum_i (C^i_{Det})_{Norm} / \#topics$ where the sum is over topics i. A detection curve (DET curve) is computed by sweeping a threshold over the range of scores, and the minimum cost over the DET curve is identified as the *minimum detection cost* or *min DET*. The topic-weighted DET cost, or score, is dependent on both a good minimum cost over the DET curve, and a good method for selecting an operating point, which is usually implemented by selecting a threshold. A system with a very low min DET score can have a much larger topic-weighted DET score. Therefore, we focus on the minimum DET score for our experiments.

4.3 Results

We present results comparing the utility of different similarity measures and use of source-pair statistics, as well as different learners for combining the similarity measures and source-pair statistics. Our system was the best performing Link Detection system at TDT2002. We cannot compare our results with the other TDT2002 Link Detection systems because participants in TDT agree not to publish results from another site. We did not participate in TDT2001, but can compare our best system on the TDT2001 test data (TDT3unp) against the results of other TDT2001 systems (we extracted the Primary Link Detection results from slides from the TDT 2001 workshop, available (as of Mar 11, 2004) at: http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt2001/PaperPres/Nist-pres/NIST-presentation-v6_files/v3_document.htm. These results are shown in Table 2. The minimum cost for our system, E, is less (better) than that of the TDT2001 systems. For this comparison, we set the bias parameter to 0.2, which reflects the expected number of linked stories computed from the training data. For the following comparative results, we set the bias parameter to reflect the probability of a linked story specified in the task definition (TDT2002), which resulted in a somewhat higher cost.

Table 3: Topic-weighted Min Detection Cost: Combined Similarity Measures and Source-Pair Specific Information (baseline system performance shown in bold)

similarity measures used	min DET Cost	
	source-pair info used?	
	no	yes
cos	0.2801	0.2532
normcos	**0.2732**	0.2533
Hel	0.3216	0.2657
Tan	0.3008	0.2748
cla	0.2706	0.2496
cos, Hel	0.2791	0.2467
normcos, cla	0.2631	0.2462
cos, normcos, cla	0.2626	0.2430
Hel, normcos, Tan	0.2714	0.2429
cos, normcos, Hel	0.2725	0.2421
cos, Hel, Tan, cla	0.2615	0.2452
cos, normcos, Hel, Tan	0.2736	*0.2418*
cos, normcos, Hel, cla	0.2614	0.2431
cos, normcos, Tan, cla	0.2623	0.2431
cos, normcos, Hel, Tan, cla	0.2608	0.2431

4.3.1 Comparison of Similarity Measures and Utility of Source-Pair Information

In this section, the effect of combining similarity metrics using an SVM and the effect of using source-pair information is examined. The results in Table 3 are divided into four sections. The upper sections show performance as measured by min DET for single similarity met-

rics and the lower sections show performance for combined similarity metrics using an SVM. The columns labeled "source-pair info used?" indicate whether source-pair specific statistics were used as input features to the SVM. The cosine, normalized cosine, Hellinger, Tanimoto, and clarity similarity measures are represented as "cos", "normcos", "Hel", "Tan", and "cla", respectively. The baseline model for comparison is the normalized cosine similarity without source-pair information (bolded), which is very similar to the most successful story link detection models (Carbonell et al., 2001; Allan et al., 2002). To assess whether the observed differences were significant, we compared models at the .005 significance level using a paired two-sided t-test where the data was randomly partitioned into 10 mutually exclusive sets.

In the upper sections, note that the clarity measure with source-pair specific statistics exhibits the best performance of the five measures, and is competitive with the combination measures in the lower right of the table. Note that the best-performing combination (italicized) did not include clarity, which may be due in part to redundancy with the other measures (Kittler et al., 1998). Compared to the normalized cosine performance of 0.2732, the improved performance of the cosine and normalized cosine measures when source-pair specific information is used (0.2532 and 0.2533, respectively; $p < .005$ for both comparisons) indicates that simple threshold normalization by the running mean is not optimal.

Comparison of the upper and lower sections of the table indicates that combination of similarity measures generally yields somewhat improved performance over single similarity link detection systems; the difference between the best upper model vs the best lower model, i.e., "cla"' vs "cos, normcos, Hel, Tan, cla" without source-pair info, and "cla" vs "cos, normcos, Hel, Tan" with source-pair info, was significant at $p < .005$ for both comparisons. And comparison of the left and right sections of the table indicates that the use of source-pair specific statistics noticeably improves performance (all models significant at $p < .005$). The lower right section of Table 3 shows a generally modest improvement over the best single metric (i.e., clarity) when using a combination of features with source-pair information.

These results indicate that although combination of similarity measures improves performance, the use of source-pair specific statistics are a larger factor in improving performance. The SVM effectively uses the the source-pair information to normalize and combine the scores. Once the scores have been normalized, for some measures there is little additional information to be gained from adding additional features, although the combination of at least two measures removes the necessity of selecting the "best" measure. For reference to the IR metrics of precision and recall, we present the results for a selected number of conditions in Table 4.

Table 4: Precision and Recall: Combined Similarity Measures and Source-Pair Specific Information

similarity measures used	source pair info used?	precision	recall
cos	no	87.45	85.33
cla	yes	87.07	88.06
cos, Hel	yes	88.83	86.75
cos, normcos, Hel, Tan	yes	88.37	87.17

Table 5: Topic-weighted Min Detection Cost: Different Learners for Combining Similarities

similarity measures used	*models*		
	voting	decision tree	SVM
cos, normcos, Hel	0.2802	0.2708	0.2421
cos, normcos, Hel, Tan	0.2810	0.2516	0.2418
cos, normcos, Hel, Tan, cla	0.2632	0.2574	0.2431

4.3.2 Comparison of Combination Models

We also investigated the use of other methods for combining similarity measures and using source-pair specific information. Table 5 compares the performance of voting, a C4.5 decision tree, and an SVM. Three sets of similarity measures were compared: 1) cosine, normalized cosine, and Hellinger, 2) cosine, normalized cosine, Hellinger, and Tanimoto (the best performing system in Table 3) and 3) the full set of similarity measures. All the SVM systems show significant improved performance at $p < .005$ over the baseline normalized cosine model, which had a cost of 0.2732 (Table 3); only one of the decision trees was significantly better at $p < .05$. The poorer performance of voting compared to the baseline may be due in part to dependencies among the different measures. None of the decision tree systems were significantly better than voting at $p < .05$; in comparison, the performance of all SVMs were significantly better than the corresponding voting and tree models at $p < .005$. The voting systems did not use any source-pair information. The decision trees used source-pair information categorically, but did not make use of source-pair statistics. The SVMs used the source-pair statistics, plus categorical source-pair information as input features. Thus, the performance of these systems tends to support the hypothesis that source-pair information, and more specifically, source-pair similarity statistics, contains useful information for the link detection task. That is, the statistics not only differentiate the source pairs, but provide additional information to the classifier.

5 Conclusions

We have presented a set of enhancements for improving story link detection over the best baseline systems. The enhancements include the combination of different similarity scores and statistical characterization of source-pair information using machine learning techniques. We observed that the use of statistical characterization of source-pair information had a larger effect in improving the performance of our system than the specific set of similarity measures used. Comparing different methods for combining similarity scores and source-pair information, we observed that simple voting did not always provide improvement over the best cosine similarity based system, decision trees tended to provide better performance, and SVMs provided the best performance of all combination methods evaluated. Our method can be used as post-processing to the methods developed by other researchers, such as topic-specific models, to create a system with even better performance. Our investigations have focused on one collection drawn from broadcast news and newswire stories in three languages; experiments on a variety of collections would allow for assessment of our results more generally.

References

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. 2000. Detections, Bounds, and Timelines: UMass and TDT-3. In *Proceedings of Topic Detection and Tracking Workshop (TDT-3)*, Vienna, Virginia.

James Allan, Victor Lavernko, and Ramesh Nallapti. 2002. UMass at TDT 2002. *Proceedings of the Topic Detection & Tracking Workshop*.

Nicholas J. Belkin, Paul B. Kantor, Edward A. Fox, and J.A. Shaw. 1995. Combining the Evidence of Multiple Query Representations for Information Retrieval. *Information Processing and Management*, **33**:3, pp. 431-448.

Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. 2002. Topic-Based Document Segmentation with Probabilistic Latent Semantic analysis. In *International conference on Information and Knowledge Management (CIKM)*, McLean, VA, pp. 211-218.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*, Wasworth International Group.

Eric Brill and Jun Wu. 1998. Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of COLING/ACL*. pp. 191-195.

Jaime Carbonell, Yiming Yang, Ralf Brown, Chun Jin and Jian Zhang. 2001. CMU TDT Report. Slides at the TDT-2001 Meeting. http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt2001/paperpres.htm (and select the CMU presentation)

Francine Chen, Ayman Farahat and Thorsten Brants. 2003. Story Link Detection and New Even Detection are Asymmetic. *Proceedings of HLT-NAACL 2003*, Companion Volume, pp. 13-15.

Christopher Cieri, David Graff, Nii Martey, and Stephanie Strassel. 2003. The TDT-3 Text and Speech Corpus. *Proceedings Topic Detection and Tracking Workshop, 2000*. http://www.itl.nist.gov/iaui/894.01/tests/tdt/research_links/index.htm

Nello Cristianini and John Shawe-Taylor. 2000. *Support Vector Machines*, Cambridge University Press, Cambridge, U.K.

W. Bruce Croft, Stephen Cronon-Townsend, and Victor Lavrenko. 2001. Relevance Feedback and Personalization: A Language Modeling Perspective. In *DELOS Workshop: Personalization and Recommender Systems in Digital Libraries*, pp. 49-54.

Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Multiple Classier Systems*, Cagliari, Italy.

Richard O. Duda and Peter E. Hart. 1973. *Pattern Classification and Scene Analysis*, John Wiley & Sons, Inc.

Tony van Gestel, Johan A.K. Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart de Moor and Joos Vandewalle. 2000. Benchmarking Least Squares Support Vector Machine Classifiers. Internal Report 00-37, ESAT-SISTA, K.U.Leuven.

Thorsten Joachims. 1999. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*, B. Schlkopf and C. Burges and A. Smola (ed.), MIT-Press.

Thorsten Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning (ECML)*, Springer, pp. 137-142.

Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. 1998. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(3), pp. 226-239.

Victor Lavrenko, James Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas. 2002. Relevance Models for Topic Detection and Tracking. In *Proceedings of HLT-2002*, San Diego, CA.

Robert E. Schapire and Yoram Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, **39**(2/3), pp. 135-168.

(TDT2002) The 2002 Topic Detection and Tracking Task Definition and Evaluation Plan http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt2002/evalplan.htm

Ian H. Witten and Eibe Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufman.