Retrieval and Novelty Detection at the Sentence Level

James Allan, Courtney Wade, and Alvaro Bolivar
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

{allan, cwade, alvarob}@cs.umass.edu

ABSTRACT

Previous research in novelty detection has focused on the task of finding novel material, given a set or stream of documents on a certain topic. This study investigates the more difficult two-part task defined by the TREC 2002 novelty track: given a topic and a group of documents relevant to that topic, 1) find the relevant sentences from the documents, and 2) find the novel sentences from the collection of relevant sentences. Our research shows that the former step appears to be the more difficult part of this task, and that the performance of novelty measures is very sensitive to the presence of non-relevant sentences.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*

General Terms

Experimentation

Keywords

TREC, novelty, redundancy, relevant sentences

1. INTRODUCTION

The goal of the TREC 2002 novelty track was to explore methods for reducing the amount of non-relevant and redundant material presented to the user of a document retrieval system. Starting from a ranked list of documents, a system's task is to first filter out all non-relevant information from those documents, reducing it to the essential components of relevance—defined in the track to be *sentences* that were relevant. As a second task, the system is required to scan those relevant sentences and discard any that do not contain new material. In the end, the user should be presented with a ranked list of sentences that contain all of the relevant information but that do not repeat information unnecessarily.

Systems participating in the track used roughly the same techniques. Relevant sentences were isolated by comparing them to the query using a vector space or language modeling framework (or something very similar). Novel sentences were then identified by comparing each sentence to all of those that occurred before it: if they were sufficiently different, they were considered novel. Broadly speaking, there were two clear conclusions of the TREC novelty track: (1) isolating relevant sentences is very difficult and (2) the value of finding novel sentences depends greatly on a system's ability to start with relevant sentences.

This paper is largely motivated by one aspect of the TREC novelty track, and of much other work on finding novel information, that has troubled us. In order to simplify the problem, researchers generally start with sets of documents that are already known to be relevant. That is, given known-relevant documents, find the documents that are novel. The assumption is presumably that the process of finding relevant material can be explored separately. Oddly, however, these efforts rarely examine what happens if that artificial assumption is lifted: what happens if the input to a system is *not* guaranteed to be relevant?

The surprising result that we have found is that the simplifying assumption may make the novelty results almost meaningless for applications as long as relevance-finding is of low quality. Specifically, if technique A is better than technique B when the input is guaranteed to be relevant, then B is sometimes superior to A when the guarantee is lifted.

In this paper we explore the task of the TREC novelty track in much greater depth than was done for the TREC workshop, with substantial focus on the problem of how novelty detection degrades as the quality of relevant information drops. In Section 3 we review the evaluation model of the TREC novelty track and describe the training material that we used to augment the handful of training topics provided in the evaluation. In Section 4 we describe the techniques and results obtained for sentence retrieval. In Section 5 we discuss the measures that we explored for finding novel sentences. The focus of this paper is in Sections 6 and 7 where we present detailed analysis of the results, including an examination of the impact of real retrieval results. We conclude in Section 8 by summarizing our findings.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.

2. RELATED WORK

A significant body of related work arises from the Topic Detection and Tracking (TDT) research and evaluation project, which is concerned with online event detection and tracking in news stories [1, 2, 5, 10, 11, 13]. However, the task approached in this work differs from TDT research in several important aspects. Most importantly, the tasks of TDT are concerned with what can be called inter-topic or inter-event novelty detection, where the concern is on whether two news stories cover the same event. In contrast, this work looks both at inter- and intra-topic novelty detection; in addition to determining whether two sentences cover the same topic, we are concerned with identifying when a sentence contains new information about that topic. Another difference is that many of the techniques developed to tackle the various tasks in TDT rely heavily upon temporal clues and other structural components specific to news reporting, information that is not guaranteed to be present in the TREC datasets used in this work. Finally, TDT is concerned with story-level online evaluation, where news stories are presented in a particular order and each one must be evaluated before the next is seen. In contrast, the task discussed in this paper is based on a batch evaluation at the sentence level.

The one task within TDT that most closely resembles this work is "new information detection" [5]. In that task, a system is expected to monitor a stream of stories on a particular topic and extract sentences that discuss new developments within the topic. That idea was addressed more satisfactorily in research on temporal summarization [3, 4], though the primary focus of their effort was to develop a useful evaluation model.

Very little research has focused on how to model intra-topic novelty. One exception to this is work on novelty detection for adaptive filtering [14] which brought together several models that had been used for other related tasks, as well as introducing new models. All five of the novelty models presented in that paper are included here as well.

Maximal Marginal Relevance (MMR) is one of the better known concepts for integrating novelty and relevance [6]. That work focused on tradeoffs between the two rather than finding the specific thresholds that are needed by the TREC novelty track's task.

This research builds upon previous work on the TREC 2002 novelty track [9, 13, 7] by presenting five additional novelty measures and analyzing the performance of all of the novelty measures on relevance results with varying levels of recall and precision.

3. TREC NOVELTY TRACK

The goal of TREC's novelty track [7] is to explore methods that reduce the amount of redundant material shown to a searcher. The task is to adapt a ranked list of documents to extract relevant sentences and then to eliminate any of those sentences that do not contain new information.

For its first year, the track's evaluation used 50 old TREC topics (taken from topics 300-450). For each topic, NIST selected only relevant documents, up to a maximum of 25. The documents were ranked using the output of a good quality manual run (system unspecified) from a past TREC evaluation. This provided a ranked list of 25 documents that had been judged on-topic (relevant).

Table 1: Statistics about the training and test data used in these experiments. All numbers are averaged over the annotated topics in each group.

Stat	NIST train	UMass train	NIST test
Count	4	48	49
Docs	18.0	23.4	22.3
Sents	1,101	1,762	1321
Rel	45.5	70.8	27.9
%Rel	4.1	4.0	2.1
Novel	42.5	57.5	25.3
%Novel	93.4	81.1	90.9

The documents were algorithmically broken into sentences and then each sentence was judged for relevance with respect to the topic. After that was done, annotators read through the relevant sentences in order and marked sentences that contained new information. Once done, some set of sentences were marked relevant, and a subset of those sentences were marked novel.

The task of a system was to replicate that sentence-level annotation by entirely automatic methods. That is, given a topic and the ordered set of relevant documents broken into sentences, a system was to find all relevant sentences and then mark some of those as novel.

To help researchers train their systems, NIST provided four training topics that were appropriately annotated. Because this was a new task and four topics provided very minimal training data, we developed an additional 48 training topics [9]. We used a method almost identical to that used by NIST, except that we hired undergraduate student to do the assessment and the documents were ranked by a simple tf·idf system rather than a manual run. Table 1 shows some statistics calculated from the training and test collections.

Evaluation in the novelty track was based on standard recall and precision measures, but applied at the sentence level, and to either relevant or to novel sentences. To minimize confusion ("relevant" sometimes means "relevant" and sometimes "novel"), we define the following special cases of recall and precision:

- *rs-recall* is the recall of relevant sentences in a set of sentences.
- *rs-precision* is the parallel measure of precision.
- *ns-recall* is the recall of novel sentences. Note that all novel sentences are by construction relevant, but not all relevant sentences are novel.
- *ns-precision* is the parallel measure of precision.

The official evaluation measures of the novelty track focused on set retrieval, evaluating the quality of the relevant set returned or the quality of the novel set returned. This led the track to adopt recall×precision as an official measure [7].[1]

In this study, however, we will focus on our ability to rank sentences by likelihood of relevance (or novelty). The TREC

[1]The official evaluation measure at the time of the TREC conference was recall×precision. However, it was later changed to the F measure.

Model	Cutoff	RS-Recall	RS-Precision
TFIDF	10%	35.57%	14.37%
	5%	23.75%	19.24%
QM	10%	34.56%	13.96%
KLD	10%	35.12%	14.19%

Table 2: RS-Recall and RS-Precision values for chosen cutoff values for the retrieved set of training sentences.

task requires identifying a cutoff threshold in that ranked list, but we generally ignore that issue here. The one exception to this is that in all experiments we had to choose a subset of the sentences, the presumed relevant set, to serve as input to our novelty algorithms. Because analysis of the training data revealed that roughly 5% of all sentences were relevant, we decided to double this number (to improve rs-recall) and assumed that the top 10% of the ranked results list for each topic was relevant. As many of our results will show, the ability to rank sentences is so poor at the moment, that it is more important to improve that capability than to find a good threshold—i.e., all thresholds are bad. Because we focus on ranking, we use rs- and ns- versions of the recall/precision tradeoff graphs, and calculate average rs- and ns-precision.

4. FINDING RELEVANT SENTENCES

For all retrieval experiments, the queries were the extended TREC topic descriptions and the items being retrieved were the sentences in the provided documents. All queries and sentences were stopped and stemmed (using the Krovetz stemmer [8]). We tried multiple retrieval models and techniques in an attempt to improve the generally poor performance, and experimented extensively with parameter tuning for each model. We experimented with three different well-known retrieval models: the vector space model with tf·idf weighting (TFIDF), a language modeling approach with the Kullback-Leibler divergence (KLD) as scoring function, and a two stage smoothing language modeling approach (QM) as described by Zhai et al. [12]. For the two language model approaches, Dirichlet and Jelinek-Mercer smoothing methods were applied.

Statistical analysis of the retrieval results for the models used shows that there is no significant difference in their performance (student's t-test p=0.05). However, we decided to use the TFIDF technique as it performed consistently (but not significantly) better than the others across different query sets. Table 2 shows the performance for chosen cutoffs and different retrieval techniques on the training set of 52 topics. These results include the use of pseudo-relevance feedback adapted to each one of the models.

4.1 Vector Space Model

In the vector space model, both the query and the sentence are represented as weighted vectors and the sentence is scored based on its distance from the query vector. In our experiments, the sentence weighting function was a form of tf·idf, the query weighting function was a variant of raw term frequency, and the distance metric was the dot product. Thus, the relevance of sentence s given query q is

$$R(s|q) = \sum_{t \in q} log(tf_{t,q} + 1) \log(tf_{t,s} + 1) \log\left(\frac{n+1}{0.5 + sf_t}\right)$$

where $tf_{t,q}$ and $tf_{t,s}$ are the number of times term t occurs in the query and sentence, respectively, sf_t is the number of sentences in which term t appears, and n is the number of sentences in the collection being scored.

4.2 Trying to Improve Performance

Given that traditional document retrieval techniques have proved unsuccessful in the task of sentence retrieval, we believe that the only way to radically improve performance is through the use of techniques specifically customized to the task. In an attempt to boost performance we tried to use known techniques such as query expansion, manual query manipulation, mixing of multiple score functions, and pseudo-relevance feedback, as well as others. Out of all of the methods tried, only pseudo-relevance feedback helped to improve performance significantly and consistently across retrieval models and data sets.

Also, extensive data analysis was executed with the objective of discovering features that would be deemed important in future sentence retrieval research. For example we tried to analyze the distribution of relevant sentences according to factors such as their position within a document in numeric or relative value or, the length of the sentence itself. We were unable to leverage that information successfully.

5. NOVELTY/REDUNDANCY MEASURES

We present seven different novelty measures. Two of them (NewWords and TREC_KL) are the measures used by UMass at TREC 2002 [9] and the other five are from Zhang et al. [14]. In all experiments, the presumed or known relevant sentences are considered in the same order in which the relevant documents were originally ranked. Multiple sentences from the same document are considered in the order in which they appear in the document. The measures described in this section are used to assign a novelty score N to each of the presumed or known relevant sentences, given the set of previously seen sentences. In keeping with the practices of Zhang et al., we treat novelty and redundancy as opposite ends of a continuous scale. Therefore ranking the sentences by increasing redundancy score is equivalent to ranking them by decreasing novelty score.

5.1 Word Count Measures

5.1.1 Simple New Word Count (NewWords)

The simple new word count novelty measure assigns each sentence a score based on the number of words it contains that have not been seen in any previous sentence. It was one of the best performing novelty measures in the TREC 2002 novelty track.

$$N_{nw}(s_i|s_1, \ldots, s_{i-1}) = \left| W_{s_i} \cap \overline{\bigcup_{j=1}^{i-1} W_{s_j}} \right|$$

W_{s_i} is the set of words contained in sentence s_i.

5.1.2 Set Difference (SetDif)

The set difference measure can be viewed as a more sophisticated version of the simple new word count that represents each sentence as a smoothed set of words. This allows for different words to carry different weights in determining the novelty score. However, set difference differs from the simple new word count in that the novelty score of sentence s_i is computed through a pairwise comparison between s_i and every previously seen relevant sentence. The previously seen sentence that is the most similar to s_i determines s_i's novelty score. In contrast, the simple new word count measure considered all of the previously seen sentences as one large set.

$$\begin{aligned} N_{sd}(s_i|s_1,\ldots,s_{i-1}) &= \min_{1\le j\le i-1} N_{sd}(s_i|s_j) \\ N_{sd}(s_i|s_j) &= \left|W_{s_i} \cap \overline{W_{s_j}}\right| \end{aligned}$$

where $w_t \in W_{s_i}$ iff $\text{count}(w_t, s_i) > k$ and $\text{count}(w_t, s_i) = \alpha_1 \cdot tf_{w_t,s_i} + \alpha_2 \cdot sf_{w_t} + \alpha_3 \cdot rsf_{w_t}$.

tf_{w_t,s_i} is the number of occurrences of word w_t in sentence s_i, sf_{w_t} is the number of presumed non-relevant sentences in the documents considered so far that contain word w_t, and rsf_{w_t} is the number of presumed relevant sentences seen so far that contain word w_t. α_1, α_2, α_3, and k are all parameters, set to different values for different collections based on the best results found in the training data.

5.1.3 Cosine Distance (CosDist)

The cosine distance metric is very common in information retrieval and has been a popular similarity measure in TDT evaluations. In the cosine distance novelty metric, each sentence is represented as a vector in m-dimensional space (where m is the number of terms in the vocabulary), and the weights on individual dimensions are determined by some weighting function. The negative of the cosine of the angle between a sentence vector and each previously seen sentence vectors then determines the novelty score for that sentence.

$$\begin{aligned} N_{cd}(s_i|s_1,\ldots,s_{i-1}) &= \min_{1\le j\le i-1} N_{cd}(s_i|s_j) \\ N_{cd}(s_i|s_j) &= -\frac{\sum_{k=1}^{m} w_k(s_i) w_k(s_j)}{\sqrt{\sum_{k=1}^{m} w_k(s_i)^2 \sum_{k=1}^{m} w_k(s_j)^2}} \end{aligned}$$

where $w_k(s_i)$ is the weight for word w_k in sentence s_i. The weighting function used in our experiments is a tf·idf function specified by the following formula

$$w_k(s_i) = \frac{tf_{w_k,s_i}}{tf_{w_k,s_i} + 0.5 + (1.5 * \frac{len(s_i)}{asl})} \cdot \frac{\log \frac{n+0.5}{sf_{w_k}}}{\log(n+1.0)}.$$

asl is the average number of words in a relevant sentence for the topic, sf_{w_k} is the number of presumed relevant sentences for the topic that contain word w_k, and n is the number of presumed relevant sentences for the topic.

Although cosine distance has performed very well as a novelty measure in past research that assigned novelty scores to full documents [14], its performance is known to degrade substantially on shorter pieces of text. Therefore, it is not expected to perform as well at the sentence level.

5.2 Language Model Measures

All of the language-model-based novelty measures presented here use the Kullback-Leibler divergence between two language models, Θ_1 and Θ_2, defined as

$$KL(\Theta_1 \parallel \Theta_2) = \sum_w p(w|\Theta_1) \log \frac{p(w|\Theta_1)}{p(w|\Theta_2)},$$

but they differ in which language models they compare.

5.2.1 Interpolated Aggregate Smoothing (TREC_KL)

Interpolated aggregate smoothing is the only language model-based novelty measure that does not perform pairwise comparisons with every previously seen sentence. Instead, a sentence is assigned a novelty score equal to the KL divergence between its language model and a single language model built on all previously scored (and presumed relevant) sentences, which is why it is referred to as an aggregate measure. The language models for both the sentence being scored and all previously seen sentences are maximum likelihood models, smoothed using linear interpolation (also known as Jelinek-Mercer smoothing).

$$N_{kl}(s_i|s_1,\ldots,s_{i-1}) = KL(\Theta_{s_i} \parallel \Theta_{s_1,\ldots,s_{i-1}})$$

where

$$\begin{aligned} p(w|\Theta_{s_i}) &= \lambda_1 p(w|\Theta_{ML_{s_i}}) \\ &\quad +(1-\lambda_1) p(w|\Theta_{ML_{s_1,\ldots,s_i}}) \\ p(w|\Theta_{s_1,\ldots,s_{i-1}}) &= \lambda_2 p(w|\Theta_{ML_{s_1,\ldots,s_{i-1}}}) + \\ &\quad (1-\lambda_2) p(w|\Theta_{ML_{s_1,\ldots,s_i}}) \end{aligned}$$

5.2.2 Dirichlet Smoothing (LMDiri)

Dirichlet smoothing of a maximum likelihood language model automatically adjusts the amount of reliance on the observed text according to the length of that text. In our experiments, this means that shorter sentences are smoothed more against the language model built on all presumed relevant sentences for that topic, whereas longer sentences are smoothed less.

$$N_{ds}(s_i|s_1,\ldots,s_{i-1}) = \min_{1\le j\le i-1} KL(\Theta_{s_i} \parallel \Theta_{s_j}) \qquad (1)$$

where both Θ_{s_i} and Θ_{s_j} are given by

$$\begin{aligned} p(w|\Theta_{s_i}) &= \frac{len(s_i)}{len(s_i)+\mu} p(w|\Theta_{ML_{s_i}}) + \\ &\quad \frac{\mu}{len(s_i)+\mu} p(w|\Theta_{ML_{s_1,\ldots,s_n}}). \end{aligned}$$

$\Theta_{ML_{s_i}}$ is a maximum likelihood language model built on sentence s_i, $\Theta_{ML_{s_1,\ldots,s_n}}$ is a maximum likelihood model built on all presumed relevant sentences for the topic, and μ is a parameter learned from training data.

5.2.3 "Shrinkage" Smoothing (LMShrink)

Shrinkage smoothing models each sentence according to the assumption that it was generated by sampling from three different language models: a sentence model, a topic model, and a model for the English language. Again, the novelty score for sentence s_i is given by equation 1, but now Θ_{s_i} and Θ_{s_j} are determined by

$$p(w|\Theta_{s_i}) = \lambda_s p(w|\Theta_{ML_{s_i}}) + \lambda_t p(w|\Theta_{ML_t}) + \lambda_e p(w|\Theta_{ML_e})$$

Θ_{ML_t} is a maximum likelihood language model of the topic and Θ_{ML_e} is a maximum likelihood language model of general English text. In our experiments, the topic model is built from the text of the extended TREC topic description.

It could also be built on the text of the presumed relevant sentences. The general English model is built on all of the sentences in the collection.

5.2.4 Sentence Core Mixture Model (LMMix)

One of the interesting properties of now popular language modeling smoothing techniques for text retrieval such as those described earlier is that they increase the probability of words that occur more in the background model(s) than in the sentence and they decrease the probability of words that occur less in the background model(s) than in the sentence. This means that some of what is different about that sentence is smoothed away, which could be an undesirable property when trying to model novelty. It may also give some indication of why such measures perform better on certain relevance results when the background model coefficient was set close to 0.

The sentence core mixture model, introduced by Zhang et al. [14] is based on an opposite assumption that words that occur more in a sentence than in the background text should have higher probability in the sentence model. The observed text is assumed to be sampled from a mixture of a "core" sentence model, a topic model, and a model of English text. However, the task of the algorithm here is to deduce the maximum likelihood sentence core model, which is then compared pairwise to each previously seen sentence core model. As with the last two measures, the novelty score is determined by equation 1 but Θ_{s_i} and Θ_{s_j} are given as

$$p(w|\Theta_{ML_{s_i}}) = \lambda_s p(w|\Theta_{s_i}) + \lambda_t p(w|\Theta_{ML_t}) + \lambda_e p(w|\Theta_{ML_e}).$$

The language model Θ_{s_i} that maximizes the likelihood of the observed sentence s_i, given fixed parameters, was computed using the technique described in Zhang et al. [15].

6. NOVELTY RESULTS

6.1 Perfect Relevance Results

Prior studies have focused on how various novelty measures perform given a collection of relevant documents. These results show how the novelty measures described in the previous section perform on the set of sentences known to be relevant to each topic.

Table 3 shows the performance of each novelty measure on the known relevant sentences for the training set of 52 TREC topics.[2] Table 5 shows the performance of each novelty measure on the known relevant sentences in the test set of 49 TREC topics.[3]

Sign tests at the 95% confidence level reveal that for both the training and testing set, there is no one novelty measure that consistently outperforms the others. However, in both cases, the set difference measure consistently performs worse than all other measures.

6.2 Best Relevance Results

Table 4 shows the performance of each novelty measure on the top 10% of the sentences in each topic from our best relevance results for the training set (36% rs-recall, 14% rs-precision) and table 6 shows the performance of each novelty measure on the top 10% of our best relevance results for the testing set (58% rs-recall, 12% rs-precision). What is most interesting is how the rankings in tables 4 and 6 are near flip-flops of the rankings in tables 3 and 5. This flip-flop effect is investigated further in section 6.3.

We were curious about whether our choice of retrieval method for finding relevant sentences would affect the relative performance of the novelty measures, so we tried running our novelty measures on the top 10% of the ranked results list produced by using a two-stage language modeling method [12] (rather than tf·idf) for retrieval. We found that the ranking of the novelty measures remained very similar for our training data.

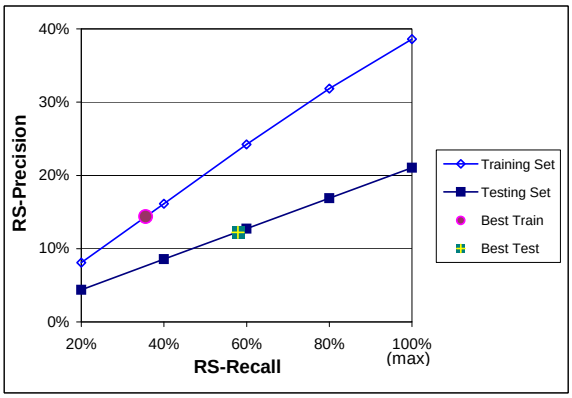

Figure 1: RS-Recall vs. RS-Precision for synthetic results with the top 10% of the total number of sentences for each topic.

6.3 Synthetic Relevance Results

Because we were intrigued by the observation that the ranking of the various novelty measures changes a great deal between the perfect relevance results and our best relevance results, we decided to construct synthetic results in order to simulate how our novelty measures would perform at different rs-recall and rs-precision levels.

We created synthetic relevance results based on the ranked list for our best relevance results. We held the number of results for each topic constant at 10% of the total number of sentences for each topic. We included the number of relevant sentences from the top of the ranked results list necessary to achieve the desired level of recall, and then filled in the remainder of the results with non-relevant sentences from the top of the ranked list. For example, suppose a topic had 10 relevant sentences out of 130 total. To build a synthetic "20% relevant" set, we select the 2 top-ranked relevant sentences. We then add the top-ranked 11 *non*-relevant sentences to yield a 10% (of 130) sample with 20% (of 10) recall. In this case the precision would be $\frac{2}{13} = 15.4\%$.

We present results from 5 different rs-recall levels: 20%, 40%, 60%, 80%, and max.[4] The max rs-recall level results represent the best performance we could have achieved using our methodology of taking the top 10% of sentences in

[2]All Random average ns-precision values presented are an average over 1000 runs.

[3]For 21 of the 49 topics in the test dataset, *all* of the relevant sentences were also judged novel which means that average ns-precision is 1.00 for all of these topics, no matter what the novelty measure - therefore these topics have no meaningful impact on the results for the known relevant testing set.

[4]For 5 of the topics in the training set, more than 10% of the sentences were relevant which means that 100% rs-recall was not possible if only 10% of the sentences were to be chosen.

Rank	Novelty Measure	Average NS-Precision
1	LMMix	0.9152
2	TREC_KL	0.9083
3	LMDiri	0.9075
4	CosDist	0.9065
5	LMShrink	0.9047
6	NewWords	0.8933
7	SetDif	0.8797
8	Random	0.8396

Table 3: Performance of novelty measures on known relevant sentences in the training set.

Rank	Novelty Measure	Average NS-Precision
1	SetDif	0.0853
2	LMMix	0.0852
3	NewWords	0.0824
4	TREC_KL	0.0769
5	LMShrink	0.0766
6	LMDiri	0.0759
7	CosDist	0.0748
8	Random	0.0693

Table 4: Performance of novelty measures on best relevance results in the training set.

Rank	Novelty Measure	Average NS-Precision
1	LMDiri	0.9581
2	CosDist	0.9574
3	TREC_KL	0.9562
4	LMShrink	0.9550
5	LMMix	0.9536
6	NewWords	0.9487
7	SetDif	0.9425
8	Random	0.9280

Table 5: Performance of novelty measures on known relevant sentences in the testing set.

Rank	Novelty Measure	Average NS-Precision
1	NewWords	0.1424
2	SetDif	0.1305
3	TREC_KL	0.1286
4	LMDiri	0.1246
5	LMMix	0.1167
6	LMShrink	0.1112
7	CosDist	0.1070
8	Random	0.0950

Table 6: Performance of novelty measures on best relevance results in the testing set.

the TREC task. In figure 1 a rs-recall/rs-precision graph is shown for these synthetic results, indicating how rs-precision also changes as rs-recall increases. The larger circle on the top line and the larger box on the lower line indicate the points where our best relevance results fall.

Figures 2 and 3 show the final ranking for the chosen novelty measures across the two different topic sets (the training and testing set). Note that the points near the bottom of the graph show better performing novelty measures. For the training set, SetDif goes from rank 1 (best) at 20% rs-recall to rank 7 (worst) at 100% rs-recall. Note that the rankings at 40% rs-recall are similar, but not identical, to the rankings seen on the best relevance results in table 4 where rs-recall was 36%. For the testing set (figure 3), there is a noticeable swap in rankings between SetDif and LMShrink. Here, note that the rankings at 60% rs-recall are similar to the rankings seen for the best relevance results in table 6 where rs-recall was 58%.

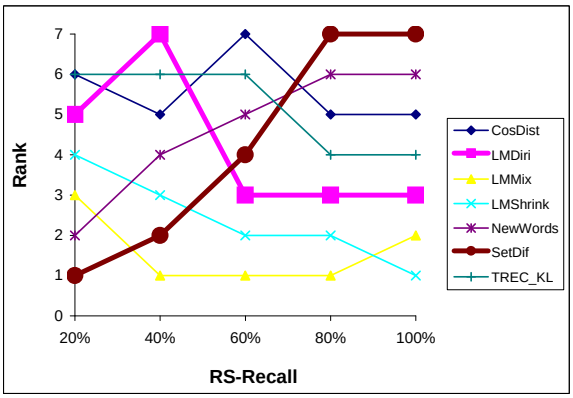

Figure 2: Ranking of novelty measures for the training set at different levels of rs-recall and rs-precision.

6.4 Summary

When the novelty detection component is handed sentences that are all relevant, the language modeling and cosine distance similarity functions work best (tables 3 and 5). However, it is the set difference measures that excel when the density of relevant sentences drops because of errors in the relevance-finding step (tables 4 and 6). The results of the experiments with synthetic data suggest that the ordering changes dramatically when recall is in the 60-80% range and precision is in the 15-30% range (see Figure 1). The cosine measure is the only measure that degrades smoothly as recall and precision decrease.

The difference between the two groups of measures is that one just counts words and the other looks at the distribution of words. When non-relevant sentences are added, the probability distribution of vocabulary shifts so that arriving sentences have more and more dissimilar distributions, suggesting that they are novel—that is, they look new because they are different from the non-relevant sentences.

On the other hand, word counting approaches are less distracted by the new words. Relevant sentences that are not novel will generally reuse vocabulary from earlier relevant sentences, and will not be sidetracked by the random vocabulary introduced by the non-relevant sentences. If the sentences are all relevant, the confusion caused by non-relevant vocabulary will disappear and all approaches should perform similarly. That is indeed what happens.

The implication of these results is that we expect that as the density of relevant documents drops even further, we anticipate that the word counting measures will continue to perform the best.

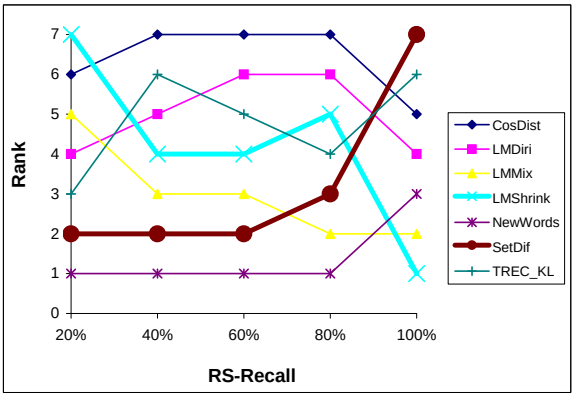

Figure 3: Ranking of novelty measures for the testing set at different levels of rs-recall and rs-precision.

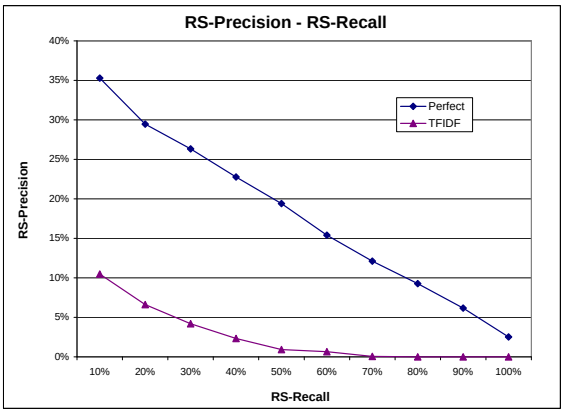

Figure 4: Interpolated rs-recall vs. rs-precision for the top 25 documents for each topic in the training set retrieved automatically.

7. REAL IR RESULTS

To explore the hypothesis of the previous section, we examine what happens when we lift the assumption that the documents with which we started are relevant. This time, instead of taking the top ranked *relevant* documents returned by a retrieval engine—the procedure used to construct the training and test sets—we took the top 25 documents returned by a retrieval system. Figure 4 shows the substantial drop in performance for finding relevant sentences when this change is made.

Because several very long documents ended up in the top 25 for the 52 topics, a total of 2,472,862 sentences were retrieved. Knowing that only 3,582 of these were relevant, it seemed unreasonable to pass the top 10% of the relevance rankings on to our novelty detection system. Instead we decided to take the same number of sentences from each topic that we used previously.

Table 7 shows the results from the novelty runs on the top-ranked sentences where rs-precision was 3.4% and the rs-recall was 8.4%. Somewhat surprisingly, NewWords and SetDif, which were formerly the best performers on low rs-precision and low rs-recall sets of sentences are the worst performers here.

This result is totally counter to the intuition expressed previously. We hypothesize that the problem occurs because the proportion of relevant sentences is now *so* low (8.4% rs-recall vs. 36% rs-recall for the best relevance run). Most of those non-relevant sentences contain new words, so end up highly ranked (incorrectly) in terms of novelty. That is, the score is now dominated by non-novel sentences that are ranked high rather than by novel sentences being ranked low. We hope eventually to extend our synthetic results analysis to much smaller proportions of relevant sentences to understand the issue better.

Rank	Novelty Measure	Average NS-Precision
1	TREC_KL	0.0138
2	LMMix	0.0135
3	CosDist	0.0129
3	LMDiri	0.0129
3	LMShrink	0.0129
6	NewWords	0.0116
7	SetDif	0.0107

Table 7: Performance of novelty measures on tf·idf relevance results using an information retrieval system to find the original documents.

8. CONCLUSION

We have presented the results of our attempts to identify relevant and novel sentences in a ranked list of relevant documents using many different methods. In our collections, finding relevant sentences has proved very difficult given the very low prior probability of relevance. This presents an interesting quandary in trying to find novel sentences because our preliminary finding from system results and synthetic results is that many novelty measures are very sensitive to the quality of the relevance results. This may be the case because certain novelty measures are more likely to flag non-relevant sentences as novel.

Few efforts have been made to model novelty explicitly–most attempts to measure novelty tend to fall back on established document retrieval techniques. Although these measures seem to work well at times, one of the more consistent novelty measures we saw here, the sentence core mixture model, also happens to be one of the measures that was not developed originally for document retrieval. However, clearly the largest hurdle remains the challenge of retrieving relevant sentences.

9. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, the Advanced Research and Development Activity under contract number MDA904-01-C-0984 and SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

10. REFERENCES

[1] J. Allan. *Topic detection and tracking: Event-based information organization*, chapter Introduction to topic detection and tracking, pages 1–16. Kluwer Academic Publishers, 2002.

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[3] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of new topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18. ACM Press, 2001.

[4] J. Allan, R. Gupta, and V. Khandewal. Topic models for summarizing novelty. In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, pages 66–71, 2001.

[5] J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. Lavrenko, R. Hoberman, and D. Caputo. Topic-based novelty detection: 1999 summer workshop at CLSP. Final Report available online at http://www.clsp.jhu.edu/, 1999.

[6] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producting summaries. In *Proceedings of SIGIR*, pages 335–336, 1998.

[7] D. Harman. Overview of the TREC 2002 novelty track. In *Proceedings of TREC 2002 (Notebook)*, pages 17–28, 2002.

[8] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202. ACM Press, 1993.

[9] L. S. Larkey, J. Allan, M. E. Connell, A. Bolivar, and C. Wade. UMass at TREC 2002: Cross language and novelty tracks. To appear in the Proceedings of the Eleventh Text Retrieval Conference (TREC 2002), 2003.

[10] M. Spitters and W. Kraaij. TNO at TDT2001: Language model-based topic detection. In *Topic Detection and Tracking Workshop Report*, 2001.

[11] N. Stokes and J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 424–425. ACM Press, 2001.

[12] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56. ACM Press, 2002.

[13] M. Zhang, R. Song, C. Lin, S. Ma, Z. Jiang, Y. Jin, Y. Liu, L. Zhao, and S. Ma. Expansion-based technologies in finding relevant and new information: THU TREC2002 novelty track experiments. To appear in the Proceedings of the Eleventh Text Retrieval Conference (TREC 2002), 2003.

[14] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual ACM SIGIR conference on research and development in information retrieval*, pages 81–88. ACM Press, 2002.

[15] Y. Zhang, W. Xu, and J. Callan. Exact maximum likelihood estimation for word mixtures. Text Learning Workshop at the International Conference on Machine Learning (ICML), 2002.