Machine Learning and Text Segmentation in Novelty Detection

Barry Schiffman and Kathleen R. McKeown
Columbia University
{bschiff,kathy}@cs.columbia.edu

Abstract

This paper explores a combination of machine learning, approximate text segmentation and a vector-space model to distinguish novel information from repeated information. In experiments with the data from the Novelty Track at the Text Retrieval Conference, we show improvements over a variety of approaches, in particular in raising precision scores on this data, while maintaining a reasonable amount of recall.

1 Introduction

The novelty detection problem seeks an automatic means of determining whether a document contains any new information on a given topic. It is a recent area of inquiry in the Natural Language Processing and Information Retrieval communities and has been explored at the last two meetings of the Text Retrieval Conference (TREC) in the Novelty Track.

At the recent TREC, the organizers at the National Institute of Standards and Tchnology (NIST) separated the track into four tasks, two of which combined passage retrieval and novelty filtering and two which concentrated on novelty filtering[1]. We chose to focus on the novelty detection, and specifically on Task Two: Given an ordered set of sentences relevant to a topic and the documents they are drawn from, choose all the "new" information – that is the information that has not appeared previously in the set of sentences [9]. Task Four is similar, but it allows the systems to see the novel sentences from the first five documents. In Tasks One and Three, systems are given only the topic statements and the source documents.

Both the retrieval and filtering subproblems are quite difficult in themselves, and it is problematic to join them and force the filtering systems to use the experimental output of the retrieval systems. The noisy input clouds what can be learned about determining novelty. Of course, someone who is building a system for the public today would have to cope with the degree of noise

[1]We were not able to participate in the Novelty Track but conducted the experiments described here after the TREC meeting in November

produced by an imperfect retrieval mechanism, but our aim is to explore the requirements of novelty detection in and of itself.

Our exploration is motivated by three intuitions. The first is that we need the original context of the sentences we are processing, and the second is that some classes of entities might be better predictors of novelty than others, and third is that a combination of strategies might complement each other, and achieve more useful results that any one on its own. A combination approach is especially appealing if the components are easy enough to be reliable.

To that end, we constructed two modules, one that uses a named entity recognizer to annotate the documents and then scans the original document to locate segments of new information of one sentence or longer. This module is tuned by a machine learning algorithm to find effective weights for various classes of entities and thresholds for finding segment boundaries. It was able to achieve high precision scores. The other module performs a pairwise comparison of the relevant sentences using a vector-space model based on the words in the sentences. It produced higher recall scores.

Our overall goal was to improve precision. It seemed from the experiences of the participants at TREC and from our own work that precision was extremely difficult to increase beyond 0.80 although 66% of the relevant sentences were novel. The first module alone succeeded in raising precision scores, but at relatively low rates of recall. After combining its results with those of the vector-space module, we raised recall to more useful levels.

The next section will review related work. Section 3 will describe the system, and Section 4 will discuss our experiments.

2 Related Work

Much of the work in this area has been done for the Novelty Track. A number of groups experimented with matrix-based methods. The group from the University of Maryland and the Center for Computing Sciences there used three techniques that operate on term-sentence matrices, QR decomposition, pivoted QR decomposition: QR algorithm, and singluar value decomposition [3]. The University of Maryland, Baltimore County, worked with clustering algorithms and singular value decomposition in sentence-sentence similarity matrices [7].

Topic words were used to cluster candidate sentences by the information retrieval group at Tsinghua University [13]. The clusters then restrict the word overlap comparisons to reduce redundancy.

The Institute of Computing Technology, the Chinese Academy of Sciences, experimented with varying the number of novel sentences according to the ordering of the source documents. In addition, they tried maximal marginal relevance, and word overlap, and found that word overlap was the most effective [10].

Meiji University embellished pairwise similarity calculations with co-occurrence data from a background corpus. It restricted the novelty comparisons to a time window for the publication dates and included an idf term in computing the sen-

tence score [12]. The national University of Taiwan also used term expansion to inform sentence similarity measures [11].

The Univerity of Iowa based its novelty decisions on a count of new named entities and noun phrases in a sentence [4].

An interesting approach at TREC 2002 was done by a group at CMU[2], which used WordNet to identify synonyms and a graph-matching algorithm to compute similar structure between sentences.

Using the TREC 2002 data, Allan [1] has done a study comparing a number of sentence-based models ranging in complexity from a count of new words and cosine distance, to a variety of sophisticated models based on KL divergence with different smoothing strategies and a "core mixture model" that considers the distribution of the words in the sentence with the distributions in a topic model and a general English model.

Our system is closest to the Iowa system since it pays a large amount of attention to a count of new named entities and noun phrases, but we give different weights to different types of named entities. We also calculate the weights of common nouns with respect to their frequency in a large background corpus and in the document set for the current topic, as does Allan's core mixture model.

3 System

This section will introduce the general outline of the system. The major components will be detailed in the subsections below.

Our system was tailored to the problem posed in the Task Two of the TREC Novelty Track. For each topic in the task, participants were given a set of sentences that have been judged relevant to the topic and were required to return a new list that contains no duplicated information. The sentences were all drawn from a set of relevant documents, 25 for each topic. There were 50 topics in the evaluation.

Our chief intuition about the problem is that the sentences have to be judged in their original context in order to achieve high precision. In a typical discourse, a segment might be introduced with sentence composed of words that clearly distinguish the topic from previous topics in the discourse, but the sentences that follow immediately after are likely to use shorthand references, such as pronouns, to realize the entities in the introductory sentence. These subsequent sentences can be hard to compare to sentences from the previous documents if the references are left unresolved.

An analysis by the TREC organizers at NIST suggests that a system should look at consecutive sentences. They determined that 84% of the relevant sentences were immediately adjacent to another relevant sentence and that the average length of a run of relevant sentences was 4.252 [9]. We examined the runs of both relevant and novel sentences, and found the same pattern. Among the novel sentences, we counted 860 runs from length 2 to length 5, accounting for 2,565 sentences, in addition to the 702 singletons. In all, there were 15,557

sentences determined to be relevant, of which 10,226 were considered novel.

This circumstance poses a dilemma. A pairwise comparison of the original sentences can fail on sentences that continue the discussion of a novel subtopic, without explicit references to the novel entities. Yet it seems to be beyond the state of the art to perform a deep analysis, like anaphora resolution, of all the documents in this task. Our solution was to utilize a surface analysis of the sentences in their original contexts, marking named entities, common nouns and verbs, using a named entity recognizer and part-of-speach tagger, and a chunker to locate noun phrases and prepositional phrases. After this was done, we scanned the sentences in the document sets, building tables of terms that were previously seen. A sentence with a number of terms that were previously unseen – or new – was considered novel. At the TREC meeting, the group from the University of Iowa [4] had the highest-precision submission using just counts of named entities and nouns. We elaborated on this approach in several ways, using the named entity recognizer in a way that provides reasonably accurate cross-document coreference, separating classes of named entities and using separate thresholds for each class, people, organizations, locations, undetermined names, common nouns, cash amounts, and verbs.

Some sentences that are not rich in such discriminating words continue a discussion of a subtopic from the previous sentence. We looked for these by examining their contexts in the original documents and tried to link them. We kept track of the current focus of the discourse, loosely following centering principles [5, 6]. When we encountered a sentence rich in terms that we could identify as either new or old, we updated the current focus accordingly. Separate thresholds were used to identify shifts to new material and returns to previously seen material. In that way, we tried to handle these sentences that did not clearly indicate if they were new or old on their own. For example, if we found a personal pronoun at the beginning of the main thought. we tried to follow the established focus. We use the chunker output here to locate sentences that begin with prepositional phrases so that we could skip them and examine the main subject. This amounted to an on-the-fly segmentation of text into new and old segments in linear time.

After we identified all the novel sections of the documents, we rescan them, using the list of relevant sentences as an oracle that tells us which were judged relevant, and therefore should be kept. And we discarded the rest.

We used a greedy, hill-climbing algorithm to determine effective values. In all, we have 11 of weights for the nominal classes and thresholds for segmentation, creating a potential search space of millions of configurations. Our learner starts with a randomly selected set of values. It chooses the next weight to update randomly, keeping changes that do not harm the score, discarding those that diminish it. Our evaluation function is the TREC score, the F-measure combination of precision and recall.

3.1 Document Analysis

We used the Talent tool from IBM [8] for sentence boundaries, part-of-speech tagging, word lemmas, named-entity recognition. By concatenating the input documents into a single file, we have Talent perform cross-document coreference. This way we got a single identifier for each named entity. The tagged documents were then fed into a finite state transducer which located the phrase boundaries.

Talent identifies people, organizations and locations, and labels others as "names".

In addition, common nouns are also valued by a score combining the document frequencies from a large background corpus with the document frequencies in the topic set. For the background, corpus, we used all the New York Times articles from 1998, 1999 and 2000 that were in the AQUAINT data. We counted the uninflected lemmas to combine the obvious morphological variations. We use a log scale for the document frequencies to create broad categories. The score is the product of the two values:

$$W = (1 - (\frac{1}{log(df_{set})}))(\frac{1}{(log(int(df_{background})))})$$

Thus a strong presence in the current document set would get added value, but not enough to outweigh the second term in the equation above, which would be near 0 for the most common words.

3.2 Segmentation

We made use of the part-of-speech tags and phrasal boundaries we located in the input texts to determine when the focus of the discourse shifts, and thus approximate topical boundaries within a document. The segments in this case were labeled as either *novel* or *old*. If we encountered long novel passages or long old passages, we made no attempt to find more subtle subtopical boundaries. We were only interested in distinguishing between new and old. Although our method was inspired by centering theory, which describes how topical shifts are signaled in a discourse, a full automation of centering principles is currently out of reach. But we made our approximation by examining the sequences of noun phrases in a sentence, imposing three tests on each sentence.

1. We begin by checking if the weight of the novel content words (including named entities) exceeds a threshold, *tnovel*. If it does, the sentence is considered novel. If the previous focus was old, this indicates the focus has shifted to a novel segment.

2. If novel words do not exceed *tnovel*, we examine the weight of the already-seen content words against a separate threshold, *told*. If they do, the sentence is considered old. If the previous focus was novel, this means the focus has shifted to an old segment.

3. The next test is threefold:

(a) If the sum of old content words and novel content words is below a threshold, *tkeep*, we assume the prior focus, novel or old, is kept.

(b) If the first noun phrase that is not contained in a prepositional phrase is a third person personal pronoun, we assume the prior focus, novel or old is kept.

(c) If none of the tests above are triggered, a second test for old content is applied, and if the value exceeds a secondary threshold, $tshift$, a novel focus is shifted to old.

4. The default is to continue the focus, whether novel or old.

The idea is to make the easier decisions first. The ordering of the tests was determined experimentally.

3.3 Machine Learning

We opted for a hill-climbing approach to find effective parameters for the system. These parameters can be divided into two kinds: the weights on the classes of words, such as locations, and the thresholds for deciding if enough of the content is novel. These values interact with each other dynamically. The decision on novelty for sentence S_i not only depends on the weights for the words it contains, but on the decision made for the previous sentence, S_{i-1}, and possible further back.

The learner (see Figure 1) is similar to a neural network where only one weight is altered at a time. If the change does not hurt results, it is accepted, otherwise the program backtracks and chooses another weight to update. At first, we required the new configuration to produce a score greater than the previous one before we accepted it. But we altered this to accept configurations that produce scores equal to the previous one. The choice of which weight to update is made at random, in an effort to avoid local minima in the search space, but with an important restriction: the previous n choices are kept in a history list and are offlimits. This list is updated at each iteration.

The configurations usually converge well within 100 iterations. We experimented with ways to initialize the starting values. We first tried handpicked values and then uniform weights, but found convergence was usually faster with random starting values.

The biggest problem was to find a way to deal with the large percentage of novel sentences. About 65% of the instances are positive, so that a random system achieves a relatively high F-measure by increasing the number of sentences it calls novel – until recall reaches 1.0. At the other extreme, a system that exclusively chose the sentences in the first document would achieve a high recall – more than 90% of the relevant sentences in the first document for each topic were considered novel.

In the Novelty Track the F-measure was set to give equal weight to precision and recall, but we wanted to be able to coax the learner to give greater weight to either precision or by adjusting the F-measure computation:

Figure 1: The learning algorithm uses a randomized hill climbing approach with backtracking

1. Initialize weights, history
   Weights take random values
2. Run the system using current weight set
3. If current score >= previous best
   Update previous best
4. Otherwise
   Undo move
5. Update history
6. Choose next weight to change
7. Go to step 2

$$F = \frac{1}{\frac{\beta}{prec} + \frac{(1-\beta)}{recall}}$$

β is a number between 0 and 1. The closer it gets to 1, the more the formula favors precision.

The design was motivated by the need to explore the problem more fully and inform the algorithm for deciding novelty as much as to find optimal parameters for the values. Thus we wanted to be able to record all the steps the learner made through the search space, and to save the intermediate states.

3.4 Vector-Space Module

Our vector-space module, which assigned all non-stop-words a value of 1, and used the cosine distance metric to compute similarity. We classified a sentence as similar to another if its cosine score exceeded some threshold, T.

$$Cos(u, v) = \frac{u \cdot v}{\|u\| \cdot \|v\|}$$

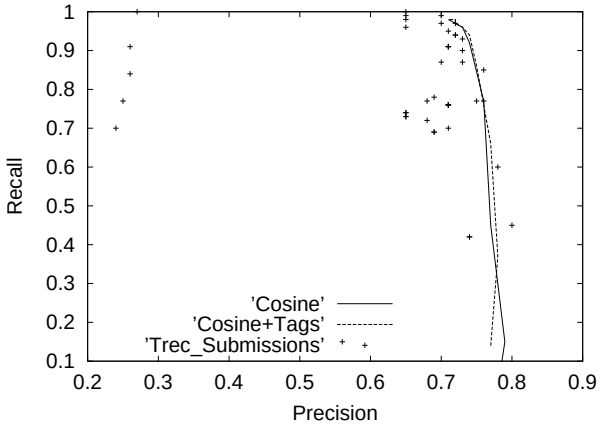
and

$$Novel(s_i) \begin{cases} true & if Cos(s_i, s_j) < T, for j = 1 \ldots i-1 \\ false & otherwise \end{cases}$$

If a sentence failed to be similar to any of the sentences previously seen, we classified as novel.

When we set T at .9, we found that we had a precision of .71 and a recall of 0.98, indicating that about 6% of the sentences were quite similar to some preceding sentence (See Figure 2). After that, each point of precision was very

Figure 2: The dots are the performance of all the submissions at TREC. The solid line shows the performance of our baseline unweighted vector-space module with a list of stop words, and the dotted line the same system using part-of-speech tags.



costly in terms of recall. Our experience was mirrored by the participants at TREC.

In practice, the range of recall was much greater than precision. Judging from the experiences of the participants at TREC and our own exploratory experiments, it is difficult to push precision above 0.80.

4 Experiments

We decided to use only the 2003 Novelty Track data. NIST changed the source and type of data, and altered both the way the topics were presented and the judgments that were made, compared with the 2002 Novelty Track. While the genre remained news, the source was changed from the last two TREC collections to the AQUAINT collection. In addition, the topics were divided between opinion and event types in 2003. The ordering of the documents was changed so that they were presented in chronological order, instead by relevance to the topic.

We divided the data into a training set of 25 topics and a testing set of 25 topics, in such a way to preserve the proportion of 56% events and 44% opinions. Our training topics had a total of 8,090 relevant sentences and 5,490 new sentences, and our testing topics, had 7,467 sentences and 4,736 new ones. The proportions of novel to relevant of 67.8% for the training set and 63.4% for the testing set are close to the combined proportion of 65.7%.

Before testing, we made several initial runs to observe the learner on the

training data only, we made several decisions about the learning procedure and one substantial change to the novelty algorithm.

With respect to the learner, we decided to use random values for the initial set of weights, instead of handpicked values or some uniform value, and to allow the program to choose these anew for each run. That way we got more insight into the behavior of the evaluation function.

At first, we set the adjustments to the weights to 0.1 increases only, but later increased the adjustment to 0.25. We allowed the adjustment by this amount to increase or decrease, a decision made randomly. The choice of weight to receive the increment or decrement is also made at random. Because the algorithm is greedy, we wanted to dampen the tendency for the program to push a particular weight too fast. We restricted the choice of the next weight by prohibiting the selection of any weight changing in the last n moves. For the final experiments we set n to 3.

We began by backtracking from any changed that failed to improve the previous score, but the results were prone to falling into local minima. Later, we altered the policy to accept any change that at least equalled the previous best score. Over all we saw a reduction of only a few points when we applied the configurations learned on the training sets to the testing sets (See Figure 4). The figure also shows the backtracking that occurs, especially toward the area of convergence.

The most immediate problem facing the learner was the large proportion of positive examples. The learner could be set to search for either the best precision or the best recall. Recall searches invariably turned out to be trivial since the system converged on configurations that simply classified a large number of examples as novel. Precision searches were better as they found configurations that achieved precision rates of more than 0.9, but at such low recall to be of little value. We then returned to using the F-measure as an evaluation function, but varying the β weight. With β weights of 0.8 to 0.97, we were able to find configurations that produced results at higher precisions than any of the participants in the 2003 Novelty Track (See Figure 3).

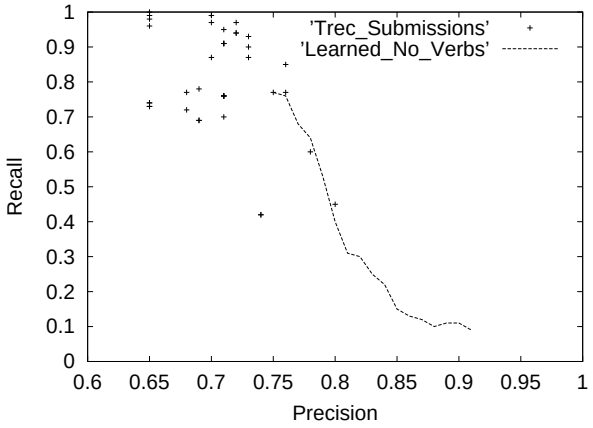
Along the way, we zeroed out the weight for verbs. At first, the inclusion of verbs seemed promising. The initial configurations seemed to start off at higher precisions, but they didn't produce any gain by the end of the learning runs. (See Figure 5).

At this point, we added the vector-space results, computed in parallel, reasoning that different approaches that produced high recall results might combine to achieve higher precision without deterioration in recall. The intersection of these two systems might be considerably better than either of the components.

Our vector-space module could achieve arbitrary high recall rates, with precision consistently above random. It operated completely on the basis of surface analysis, using only the words in the documents. It however encountered a relatively low ceiling on precision, as seen in Figure 2, dropping straight down around 0.78.

To make the combination work, we needed higher recall scores from the segmentation module. So we began reducing the β values from 0.8 to 0.6 and then

Figure 3: Comparing the segmentation module with learned weights against the submissions at the TREC meeting.



to 0.5, but this time were interested in the configurations that were discovered earlier in the learning search, those with moderate precision and recall scores. By 100 iterations, these searches would often converge to a configuration of weights that produced a precision near random, and a recall near perfect, but earlier iterations on the testing sets often produced relatively high recall at precisions above 0.75. By themselves, these were similar to several of the stronger submissions in the Novelty Track.

But when we combined the two modules by taking intersections of their selection, we saw substantial improvements in results (See Figure 6).The best combinations can be seen in Table 1, giving the β values for the learner, the cosine similarity threshold, the iteration when the configuration was found, and precision and recall.

To illustrate the way the combination works, see row 5 in Table 1. This was the 36th iteration for the learner, and alone on the test set, the configuration produced a precision score of 0.78 and a recall of 0.62. It is paired with the vector-space system at a 0.40 similarity theshold, which itself produced a precision score of 0.75 and a recall of 0.86. The intersection operation removed many of the inaccuracies of both systems, resulting in a precision of 0.80, higher than either system alone, while reducing the recall from a maximum of 0.62 to 0.54. Note that the learner began its $\beta = 0.50$ run at a particularly high precision, with a precision of 0.83 and 0.28. Ordinarily the random configurations start at a somewhat lower precision and a somewhat higher recall.

In this run, in which the β value for the learner was set at 0.50, we obtained useful configuration through 49 iterations, but the learner jumped to a configuration of 0.63 precision and 0.99 recall – presumably because of the negative *nov* (novelty) weight, which would have the effect of accepting all sentences as

Figure 4: Showing the difference between the training and testing for the segmentation module.

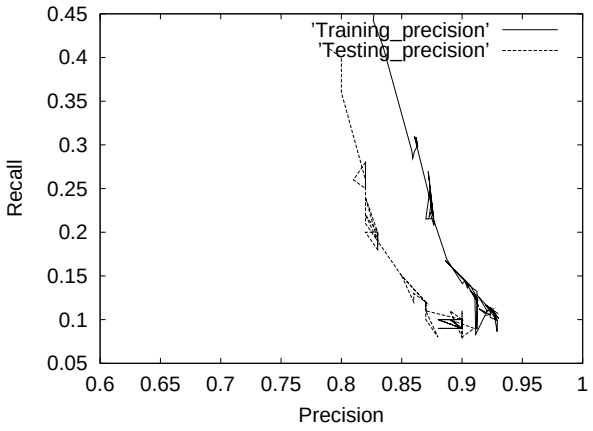


Figure 5: The effect of using verbs or not

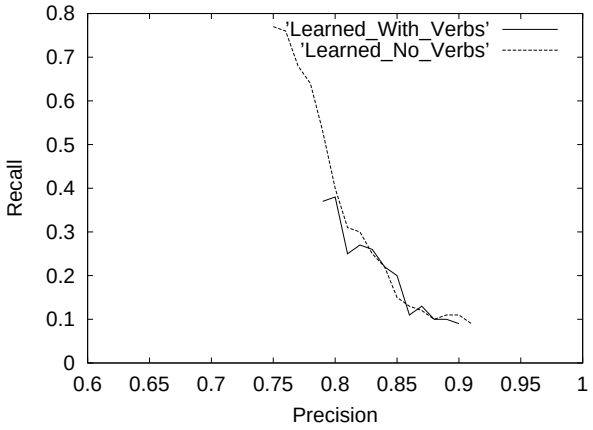


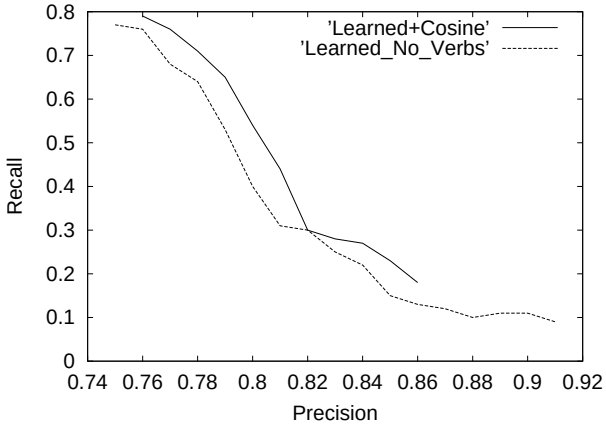
Table 1: The table shows the best of the combination results for two runs of the learner with the vector-space module. F-bias refers to the weight given to precision in balancing precision and recall to compute the F-measure. Iter identifies which iteration in the run of the learner the result arose from. The Cos value is the cosine similarity threshold. Higher values mean fewer sentences are similar, and thus more are accepted as novel. In all cases, the combination is done by intersection of the two sets of answers.

F-bias	Cos.	Iter.	Prec.	Recall
0.50	0.70	77	0.76	0.77
0.60	0.50	99	0.77	0.76
0.60	0.40	99	0.78	0.71
0.50	0.40	46	0.79	0.64
0.50	**0.40**	**36**	**0.80**	**0.54**
0.50	0.30	36	0.81	0.44
0.50	0.30	10	0.82	0.30
0.50	0.50	5	0.83	0.28
0.50	0.50	1	0.84	0.27
0.50	0.50	2	0.85	0.23
0.50	0.30	2	0.86	0.18

Table 2: This table gives some example configurations generated by the learner. The *Start* column shows the random initial values, the *Actual* column shows the values obtained in the 36th iteration, and the *Final* column gives the end configuration. The *nov* weight is the primary novelty threshold, and the *old* weight is separate test to determine if a sentence is definitely old. The *minshift* and *minkeep* weights are secondary tests on whether to continue or drop a novel focus for sentences that do not clearly contain many discriminatory words. The remaining weights are for classes of types of nouns and of verbs. This is the run that contributed to the combination result shown in Table 1

Key	Start	Actual	Final
nov	0.950	0.450	-0.050
old	0.953	1.453	2.453
minshift	0.584	1.084	2.333
minkeep	0.269	0.519	0.269
loc	0.734	1.234	1.734
org	0.355	0.856	1.356
name	0.457	0.457	1.707
cash	0.531	0.781	0.281
hum	0.919	1.419	1.169
noun	0.975	1.975	2.475
vrb	0	0	0

Figure 6: The chart shows the benefit of combining the learned scores with the vector-space model. The combination is done by taking the intersection of the sentences labeled as novel by both modules.



novel (See Table 2. Since the evaluation function weighted precision and recall evenly, this was an impossible F-measure to improve upon.

5 Conclusion

We composed a system that combines the output of one module that produced higher precision scores with another that reached higher recall scores to surpass previous results in the TREC Novelty Track. Both modules relied on surface analysis of the documents and offer an efficient solution to the problem.

The module that was better on precision uses the original context of the sentences, and machine learning to find the relative weights for different classes of entities. Its output was combined with a more traditional vector-space model. Together, the combination of different types of evidence for novelty is promising and suggests that further gains could be made by adding other classifiers.

Our study of the data and our experiments have given us many interesting insights into the problem. A completely naïve approach can produce a competitive score, but the relatively high F score is produced by returning a very large percentage of the sentences. It seems that brevity deserves a premium here. Some measurement of the relative importance of the passages would greatly enhance the utility of the system and we would also like to look at ways to factor in the importance of our selections.

References

[1] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the ACM SIGIR conference on research and development in information retrieval*, 2003.

[2] K. Collins-Thompson, P. Ogilvie, Y. Zhang, and J. Callan. Information filtering, novelty detection and named-page finding. In *Proceedings of the 11th Text Retrieval Conference*, 2002.

[3] J. M. Conroy, D. M. Dunlavy, and D. P. O'Leary. From trec to duc to trec again. In *TREC Notebook Proceedings*, 2003.

[4] D. Eichmann, P. Srinivasan, M. Light, H. Wang, X. Y. Qiu, R. J. Arens, and A. Sehgal. Experiments in novelty, genes and questions at the university of iowa. In *TREC Notebook Proceedings*, 2003.

[5] P. C. Gordon, B. J. Grosz, and L. A. Gilliom. Pronouns, names and the centering of attention in discourse. *Cognitive Science*, 17(3):311–348, 1993.

[6] B. J. Grosz, A. K. Joshi, and S. Weinstein. Centering: A framework for modeling the local coherence of discourse. *Association for Computational Linguistics*, 21(2):203–225, 1995.

[7] S. Kallurkar, Y. Shi, R. S. Cost, C. Nicholas, A. Java, C. James, S. Rajavaram, V. Shanbhag, S. Bhatkar, and D. Ogle. Umbc at trec 12. In *TREC Notebook Proceedings*, 2003.

[8] Y. Ravin, N. Wacholder, and M. Choi. Disambiguation of proper names in text. In *Proceedings of the 17th Annual ACM-SIGIR Conference*, 1997.

[9] I. Soboroff and D. Harman. Overview of the trec 2003 novelty track. In *TREC Notebook Proceedings*, 2003.

[10] J. Sun, W. pan, H. Zhang, Z. Yang, B. Wang, G. Zhang, and X. Cheng. Trec-2003 novelty and web track at ict. In *TREC Notebook Proceedings*, 2003.

[11] M.-F. Tsai, W.-J. Hou, C.-Y. Teng, M.-H. Hsu, C. Lee, and H.-H. Chen. Similarity computation in novelty detection and generif annotation. In *TREC Notebook Proceedings*, 2003.

[12] M. University. Meiji university web and novelty track experiments at trec 2003. In *TREC Notebook Proceedings*, 2003.

[13] M. Zhang, C. Lin, Y. Liu, L. Zhao, L. Ma, and S. Ma. Thuir at trec 2003: Novelty, robust, web and hard. In *TREC Notebook Proceedings*, 2003.