Columbia University in the Novelty Track at TREC 2004

Barry Schiffman
Columbia University
New York, N.Y. 10027
bschiff@cs.columbia.edu

Kathleen R. McKeown
Columbia University
New York, N.Y. 10027
kathy@cs.columbia.edu

ABSTRACT

Our system for the Novelty Track at TREC 2004 looks beyond sentence boundaries as well as within sentences to identify novel, nonduplicative passages. It tries to identify text spans of two or more sentences that encompass mini-segments of new information. At the same time, we avoid any pairwise comparison of sentences, but rely on the presence of previously unseen terms to provide evidence of novelty. The system is guided by a number of parameters, both weights and thresholds, that are learned automatically with a randomized hill-climbing algorithm. During learning, we varied the target function to produce configurations that emphasize either precision or recall. We also implemented a straightforward vector-space model as a comparison and to test a combined approach.

1. INTRODUCTION

The novelty detection problem seeks an automatic way of identifying any new information in a document, or documents, 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 three meetings of the Text Retrieval Conference (TREC), in the Novelty Track.

After the first Novelty Track in 2002, the National Institute of Standards and Technology (NIST) separated the track into four tasks, two of which combined passage retrieval and novelty filtering and two of which concentrated on novelty filtering, giving participants the choice of whether to do the combined tasks (Tasks 1 and/or 3) or whether to focus on the novelty detection alone (Tasks 2 and/or 4). In the combined tasks, participants have to first choose the sentences that are "relevant" to a given topic from a set of documents, and then make a second pass to remove duplicates. In the pure novelty task, participants are given an ordered set of relevant sentences and must filter them to choose all those with "new" information – that is the information that has not appeared previously in the set [10].

Both the retrieval and filtering tasks 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. We did only Task 2 of the Novelty Track this year, since it is most closely related to our ongoing research into creating updates or bulletin summaries for an on-line news browsing system.

Our submission for the Novelty Track, called SumSeg, is based on observations of the data we collected for the development of our update summarizer. We saw that new information sometimes appears in passages that are two or more sentences long, and sometimes only in clauses embedded in a sentence. (Task 4 is similar to Task 2, but it allows the systems to see the novel sentences from the first five documents. Time constraints prevented us from submitting runs for it that would have made use of additional input.)

In order to recognize novelty in both cases – segments of two or more sentences, and embedded clauses that are only part of a sentence – we avoid direct sentence similarity measures, and consider previously unseen words to be the main evidence of novelty. SumSeg has a number of thresholds for deciding how much novelty is necessary to trigger a novel classification. We implemented a randomized hill climbing algorithm to learn thresholds for how many new words would trigger a novel classification. We also sought to learn different weights for different types of nouns, for example, persons, or locations or common nouns. In addition, we included a mechanism to allow sentences that had few strong content words to *continue* the classification of the previous sentence. The basic SumSeg system is described in [8]. Finally, we used two statistics, derived from analysis of the full Aquaint corpus, to eliminate low-content words.

For TREC 2004, we submitted a total five runs: the first two used learned parameters that aimed at high precision output, and the third at high recall. The fourth run was a straightforward vector-space model, with a cosine similarity metric, used as a baseline, and the fifth was a combination of the high recall run with the vector-space model. Training was done on the 2003 TREC novelty data.

Over all, we were most interested in trying 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 much beyond a random selection of relevant sentences. In the 2003 Novelty Track, the top precision was 0.80 although 66% of the relevant sentences were novel. The median precision among all 45 runs in 2003 was 0.70, and the average 0.635. If we remove the five runs by one participant that were in the 0.2 range, possibly because of some misunderstanding, the median is still only 0.71 and the average 0.687. In our summarization work, we especially value conciseness and our long-term goal would be to find the minimal output of a novelty system.

The next section will review related work. Section 3 will describe the system, and Section 4 will discuss our experiments. Finally Section **??** will preview our performance in this year's Novelty Track.

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 singular value decomposition [3]. The University of Maryland, Baltimore County, worked with clustering algorithms and singular value decomposition in sentence-sentence similarity matrices [6].

Topic words were used to cluster candidate sentences by the information retrieval group at Tsinghua University [14]. 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 by the ordering of the source documents. They also tried maximal marginal relevance, and word overlap, and found that word overlap was the most effective [11].

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 scoring sentences [13]. The national University of Taiwan also used term expansion to inform sentence similarity measures [12].

The University of Iowa based its novelty decisions on a count of new named entities and noun phrases in a sentence [5].

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] compared 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 2 of the TREC Novelty Track. For each of the 50 topics, 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 sentences that were covered by information seen earlier in the input. The relevant sentences were all drawn from a set of documents, at least 25 for each topic. Some topics had additional documents, some not relevant to the topic, that were included to increase the difficulty of the tasks, but these would have no impact on Task 2. The topics were evenly divided between opinion and events.

Our chief intuition about the problem is that contextual features are important in classifying the sentences. At first we tried to leave the sentences in their original context. This strategy incurred a considerable amount of additional processing. By limiting the input to the relevant sentences, we found that our results did not deteriorate since we had the sentence indices so the program could determine when two were adjacent or not. In a typical discourse, a segment might be introduced with sentence composed of words that clearly indicate novelty, 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 in 2003 were immediately adjacent to another relevant sentence and that the average length of a run of relevant sentences was 4.252 [10].

Table 1 shows that more than half the novel sentences at TREC 2004 appear in consecutive runs of two or more.

Length of Run	Count
1	1338
2	421
3	132
4	72
5	43
6	22
7	11
8	2
9	3
10	3
11	2
12	2
15	2
17	1

Table 1: Novelty often comes in bursts

This circumstance poses a dilemma. A pairwise comparison of 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, marking named entities, common nouns and verbs, using 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 sufficient number of terms that were previously unseen – or new – was considered novel. The thresholds were learned, as described below.

At the TREC 2003, the group from the University of Iowa [5] had the highest-precision submission using just counts of named entities and nouns. We elaborated on this approach, 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, unspecified 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 the previous sentences. 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 *novel* and those returning to *not novel.* 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 followed the established focus. We use the chunker output here to determine the probable main subjects.

We used a greedy, hill-climbing algorithm to determine effective values. In all, we have 11 features, either 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, i.e., the F-measure combination of precision and recall.

3.1 Document Analysis

We used the Talent tool from IBM [7] for sentence boundaries, part-of-speech tagging, word lemmas and 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. Talent identifies people, organizations and locations, and labels others as “names”. The tagged documents were then fed into a finite state transducer that located the phrase boundaries.

In addition, common nouns are weighted 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. In revising the system for this year’s TREC evaluation, we added a new feature, *promiscuity*, which is derived from an analysis of the APW portion of the Aquaint corpus. The goal is the same as our use of document frequencies, but the method for identifying these words is their contextual distribution. The idea is that words occurring in too many different contexts will not be of much use in classifying the sentences.

The promiscuity values seek to eliminate words that are too vague to count for similarity/dissimilarity judgments. They are constructed by analyzing tables of document cooccurrences and deciding which are closely bound to a large number of other words. The base statistic used is the log likelihood ratio [4].

$$\lambda = \frac{\max_{\omega \in \Omega_0} H(w;k)}{\max_{\omega \in \Omega} H(\omega;k)},$$

where

$$H(\omega;k) = H(p;n,k) = p^k(1-p)^n - k \left(\begin{array}{c} n \\ k \end{array} \right)$$

We took the λ for each pair of words that co-occurred in documents, and then computed the mean and variance of matches for a word, plus a count of the number of significant matchups. The statistics were separated by part of speech, so the results for nouns co-occurring with other nouns was separate from those for nouns with verbs. The values represent a combination of the learning algorithms over the different categories. We used a threshold of 0.55. Here, if the value exceeds that threshold, the word is eliminated from consideration. Table 2 shows some of the rejected nouns, the middle column the verbs and the right column, adjectives.

way	1.0	use	0.75	important	1.0
use	1.0	turn	0.75	human	1.0
type	1.0	try	0.75	high	1.0
time	1.0	stay	0.75	hard	1.0
thing	1.0	show	0.75	great	1.0

Table 2: A sampling of nouns, verbs and adjectives that were found to be used in too many contexts to convey much meaning on their own.

We are not arguing that these words have no content or meaning, but that they are either intrinsically vague or are commonly used in structures that are semantically dominated by another word, like “a type of vehicle”. The word type provides information about the object, but vehicle is the word we want.

3.2 Segmentation

We made use of the part-of-speech tags and phrase boundaries 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 *not novel.* We made no attempt to find or label topical boundaries or differentiate between novel segments. We were only interested in distinguishing between new and old. In examining the sequences of noun phrases in a sentence, we imposed three tests on each sentence.

1. We begin by checking if the sum of the weights of the novel content words (including named entities) exceeds a threshold, T_{novel}. 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 T_{novel}, we examine the weight of the already-seen content words against a separate threshold, T_{old}. 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, T_{keep}, 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, T_{shift}, 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, like people or 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 possibly further back.

The learner (see Figure 1) is similar to neural networks where only one weight is altered at a time, and to genetic algorithms, where changes to the hypothesis are selected at random and evaluated. 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 off limits. This list is updated at each iteration.

```
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
```

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

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.

In training on the 2003 data, 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:

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

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.

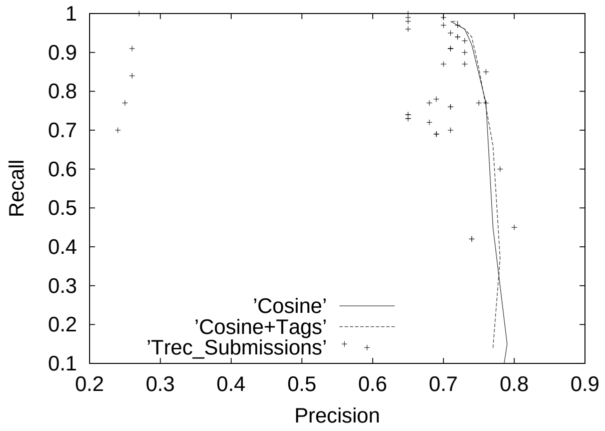

Figure 2: The dots are the performance of all the submissions at TREC. The solid line shows the performance of our baseline unweighted vector-space module with a list of stop words, and the dotted line the same system using part-of-speech tags.

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

and

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

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 costly in terms of recall. Our experience was mirrored by the participants at TREC 2003.

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.

In our initial exploration, where we wanted to see how well our hypothesis might generalize, 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 learning rate at 0.1, but later increased the adjustment to 0.25. We allowed the updated weights to increase or decrease by this amount, wit. 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, falling into local minima. 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 3). The figure also shows the backtracking that occurs, especially toward the area of convergence.

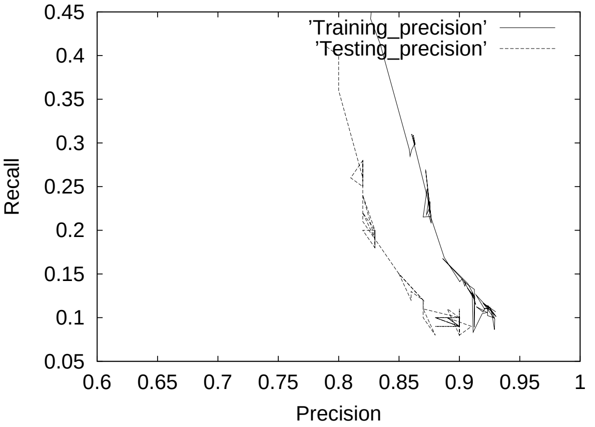

Figure 3: Showing the difference between the training and testing for the segmentation module.

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 4).

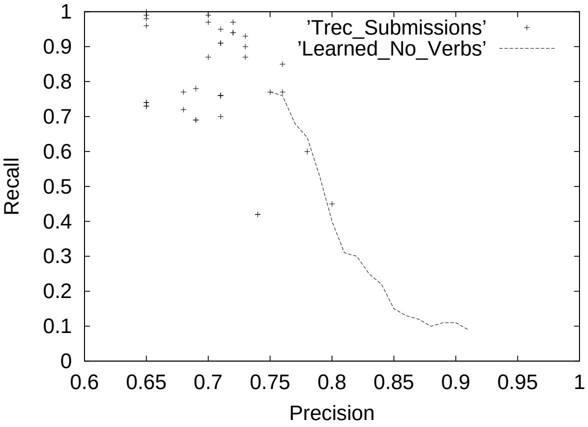

Figure 4: Comparing the segmentation module with learned weights against the submissions at the TREC meeting.

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, 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 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 5).One of the stronger combinations was to take the intersection of a recall-oriented run of *SumSeg* after 50 iterations with the vector-space model at a cosine similarity threshold of 0.40. The result achieved 0.80 precision, with 0.54 recall on the unseen test examples.

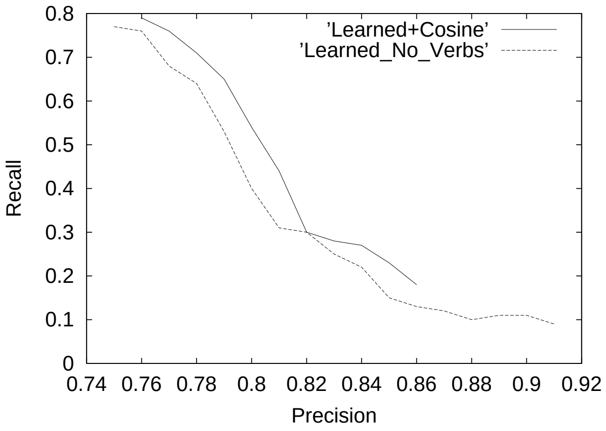

Figure 5: The chart shows the benefit of combining the learned scores with the vector-space model. The combination is done by taking the intersection of the sentences labeled as novel by both modules.

4.1 Fine Tuning

Since we added the promiscuous words features, we wanted to re-examine the results of learning, and we wanted to retrain on the all the 2003 data, but we felt we could build on the results from the earlier training. We began with several configurations learned in the first round of experiments – those that held up well on the test examples after training. We added the new feature, and allowed the novelty and focus shift thresholds to change. Since we were only dealing with the list or relevant sentences, we thought these parameters might benefit from change. Along with these, we tried restoring the verb weights, which had been zeroed out in round 1, and trying higher values for common nouns and verbs than before. The reasoning was that with the promiscuous words features, the content of the allowed words would be more reliable.

Table 3 shows the final configuration used for Trec 2004.

5. EVALUATION

Our results are encouraging, especially since the configurations that were oriented toward higher precision, indeed, achieved the best precision scores in the evaluation, with our best precision run about 20% higher in precision than the best of all non-Columbia runs (See Figure 6.) Meanwhile, our recall-oriented run was one of eight runs that were in a virtual tie for achieving the top f-measure. These eight runs were within 0.01 of one another in the measure.

Table 4 shows the numbers of our performance of our five submissions. *Prec1* had an F-score close to the average of 0.577 for all systems, while while *Prec2* was 50% ahead of a random selection in accuracy.Both our *Combo* system and our baseline *Cosine* were above average in F-measure. Our

Key	Prec1	Prec2	Recall
dfw	on	on	on
prom	on	on	on
nov	0.70	1.50	0.24
old	0.95	0.96	2.27
minshift	0.67	0.67	2.5
minkeep	1.00	1.00	2.45
loc	0.73	0.74	1.95
org	0.61	0.64	1.29
name	0.46	0.41	1.52
cash	0.53	0.80	1.98
hum	0.92	1.19	2.39
noun	1.47	2.00	2.85
vrb	1.25	3.00	1.25

Table 3: A comparison of the three SumSeg configurations for the 2004 TREC Novelty Track. Prec1 and Prec2 were two selected for ability to raise precision. In the initial training, from the first round, the fitness function for the learning algorithm favored precision over recall. For recall, the f-measure was emphasized, which tends to favor high recall systems. Note that the novelty threshold, *nov*, is relatively high in the precision-biased configurations, while the weights on classes of words are relatively low.

emphasis on precision is justified in a number of ways, although the official yardstick was the F-measure.

First, we approach the problem from the summarization task, where compression of the report is valuable. Table 4 shows the lengths of our returns. It is impossible to compare these precisely with other systems, because the averages given by NIST are averages of the scores for each of the 50 sets, and we do not have the breakdown of the numbers by set for any submissions but our own. However, we can estimate the size of the other output by considering average precision and recall as if they were computed over the total number of sentencesin all 50 sets. This computation shows an average length of about 6,500 sentences and a median of 6,981 – out of a total of 8,343 sentences. However, this total includes some amount of header material, not only the headline, but the document ID and other identifiers, the date and some shorthand messages from the wire services to its clients. In addition, a number of the sets had near perfect duplicate articles. We contend there is little value in a system that does no more than weed out some non-narrative material and very simple cases, even though they might have achieved high F-measures.

Second, our experience, and the results of other groups, shows that it is much more difficult to achieve high precision than high recall. In all three years of the Novelty Track, precision scores tended to hover in a narrow band just above what one would get by mechanically selecting *novel* for all sentences. This phenomenon was apparent in 2002 when more than 90% of the relevant sentences were novel, and in 2003 when about 65% of the relevant sentences were novel and in 2004 when only 41% were novel.

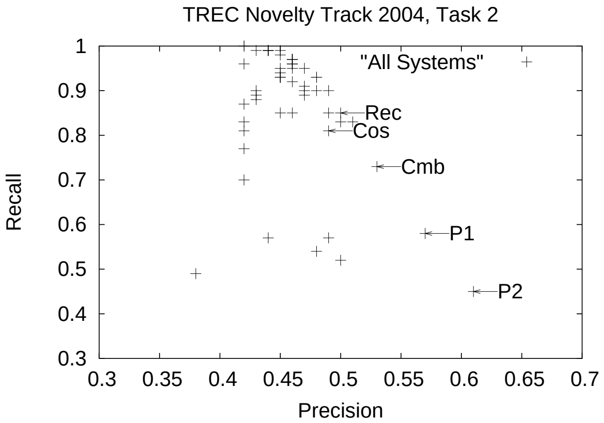

Figure 6: The graph shows all 54 submission in Task 2 for the Novelty Track, with our five submissions labeled. Our precision-oriented runs were well ahead of all others in precision, while our recall-oriented run was in a large group that reached about 0.5 precision with relatively high recall.

Finally, the F-measure is problematic in this task, as NIST concedes in its overview [9], because the same score can be achieved by vastly different systems. In cases where the targets are relatively few, accuracy and coverage are better matched in difficulty.

Run-Id	Precision	Recall	F-meas	Length
Prec1	0.57	0.58	0.562	3276
Prec2	0.61	0.45	0.506	2372
Recall	0.50	0.85	0.617	5603
Cosine	0.49	0.81	0.599	5537
Combo	0.53	0.73	0.598	4578
All Nov	0.41	1.00	0.581	8343
Average	0.46	0,86	0.577	6500

Table 4: Comparison of results of Columbia's five runs, compared to a random selection of sentences, and the overall average F-scores by all 55 submissions.

A comparison with the 2003 results is difficult. In 2004, there were 8,343 relevant sentences, but only 3,454, or 41.3% were judged novel, a sharp drop from the previous year, when 65% of the relevant sentences were judged novel.

6. CONCLUSION

We built a system that combines that is capable of being tuned to emphasize either precision or recall, using machine learning to find effective parameters. Although we cannot compare the results to runs submitted by other groups, it seems that we did well. We have already incorporated some of the strategies that worked here into our larger system.

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.

The input sets in the Novelty Track have changed greatly over the three years. The proportion of novel sentences to relevant sentences has steadily dropped. The first year, more than 90% of the relevant sentences were novel; then 65% and now 41%. There is considerable variation among the sets in any one year, and we are curious about finding a way to automatically categorize sets and adjust the parameters of the classifier. Our precision systems seem to do better on the sets with a lower proportion of novel to relevant, but they are erratic in sets with higher proportions. We have found no clear difference between event and opinion sets, but that may be an additional area of inquiry.

Finally we would like to find an efficient way of trying more complete reference resolution, as that could make the notion of novel segments stronger.

7. 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] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[5] 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.

[6] 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.

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

[8] B. Schiffman and K. R. McKeown. Machine learning and text segmentation in novelty detection. Technical Report CUCS-036-04, Columbia University, 2004.

[9] I. Soboroff. Draft overview of the trec 2004 novelty track. In *The Thirteenth Text Retrieval Conference (TREC 2004) Noteboo k*, 2004.

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

[11] 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.

[12] 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.

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

[14] 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.