An Investigation Into the Detection of New Information

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

Abstract

This paper explores new-information detection, describing a strategy for filtering a stream of documents to present only information that is fresh. We focus on multi-document summarization and seek to efficiently use more linguistic information than is often seen in such systems. We experimented with our linguistic system and with a more traditional sentence-based, vector-space system and found that a combination of the two approaches boosted performance over each one alone.

1 Introduction

The voluminous amount of information now in digital form poses an important challenge – to distinguish new material from material in previously seen documents. The stream of news from around the world on the World Wide Web is but one form of this deluge of data. Data from the world financial markets, government actions, court decisions, scientific research can all be tapped, but that value will be greatly diminished if readers must sift through the same material over and over again.

A similar theme surfaces in many active areas of computational linguistics, such as question answering, information retrieval, and summarization, where systems must determine similarity between two pieces of text. For new information detection, a system must determine differences, a task made difficult because of the seemingly endless ways in which writers can realize the same content. A sentence is a recursive structure and can be composed of some number of embedded clauses. One document may have a complex sentence. An alternate document may break that sentence into two or three sentences without changing any meaning. Further, the writers may choose entirely different words to make the same points. Here is an example of the same facts presented in rather different terms, as they appeared in two news articles written at the same time about the murder charges against the former TV actor Robert Blake:

> The former "Baretta" star is accused of fatally shooting Bonny Lee Bakley, 44, nearly two years ago as she sat in his car near the Studio City restaurant where they had dined. He also is accused of soliciting two stuntmen and conspiring with Caldwell to kill her.

This second article contains this sentence that partially covers the material above.

> Blake will be tried on charges of murdering Bonny Lee Bakley, solicitation of murder, conspiracy and the special circumstance of lying in wait.

Elsewhere in the second article, the reader can find out that the shooting took place outside a Studio City restaurant, and that the stuntmen and Caldwell were involved. In effect, the syntactic pieces of the article have to be mixed and matched in order to determine that they contain the same information.

We are seeking to find a middle ground between full interpretation and pure statistical approaches for such problems. A full translation of these two sentences into a logical form that can be directly compared is not likely to scale up. A comparison of words in sentences is not likely to reach beyond the structural differences.

Our approach enriches the input texts in three ways: 1. a semantic dictionary combining manually built resources with corpus analysis to equate different expressions that convey the same information – like "fatally shooting" with "murdering", and "accused of" with "on charges of"; 2. parsing to decompose sentences into clauses in order to approximate atomic facts; 3. several kinds of reference resolution ultimately to determine that "the star" is indeed "Blake".

This paper details the current state of the system, and also describes our strategy to build a corpus of sound human judgments for the development of the system. We present a series of experiments and show that an approach combining our linguistically oriented strategy with a simple sentence-based system succeeded in lifting precision scores substantially.

2 Problem Specification

We define the new-information-detection problem in terms of multi-document summarization, where we are given a stream of documents on a particular event, and are asked to provide a summary of only the information that is new at a certain point in time. For example, suppose we are tracking the online news about the latest outbreak of a computer virus. The user may check the system on Day_n but only wants to know if there have been any developments since Day_{n-1}.

Our system will work with documents produced over a short span of days, which would reflect the burstiness of news events, where a new event receives much attention at the beginning and then gradually recedes from view. We are using documents readily available on the World Wide Web. We have been experimenting with pairs of article, to keep the task more manageable in development; in this context, the system's task is to find the new information in one article that is not in the other. However, our system can process any number of articles, and separate the articles into clusters of old (yesterday's news) and new (today's).

We paid close attention to the quality of the corpora we will use for development and testing of the system. An earlier pilot annotation of pairs of news articles convinced us of the need for precision in the markup, in keeping with the difficulty of the task. We require the people doing the annotation to reach agreement on each passage, forcing them to look closely at the texts and to make decisions on the novelty of passages that they can defend. Considerable attention has to be focused on a large number of comparisons, and the annotators have told us that the task is much harder than they anticipated.

The annotators so far have prepared 31 pairs of documents, which contain 3,732 clauses. The *new* articles contained 1,943 clauses. The annotators identified 1,214 clauses (62.5%) as containing new information. We discuss this annotation fully in Section 5.

3 Related Work

Much of the work in new information is related to the Text Retrieval Conference (TREC) Novelty Track, which is a substantially different version of the problem. In TREC, systems operate on material returned by an information retrieval engine. Once relevant documents are pulled from a collection, relevant passages are then extracted from the articles in the *Retrieval Task*, and duplicates are removed in the *Novelty Task* so that $Sentence_n$ does not echo material in $Sentence_1 ... Sentence_{n-1}$.

Since TREC 2002, 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. On the TREC 2002 test data, all the variations performed within a narrow range (most falling with 1 percentage point of each other in precision), and the authors note that no one measure outperformed the others.

Other interesting approaches at TREC 2002, included a group at CMU[4], which used WordNet to identify synonyms and a graph-matching algorithm to compute similar structure between sentences. WordNet was also used by the group from CUNY[8] with a variation on the Dice coefficient to measure sentence similarity. The University of Iowa[5] tried various combinations of weights for named entities and noun phrases with sentence and document similarity.

In all, the novelty task in 2002 was clouded by its dependence on the relevance task, where all the sentences relevant to a topic had to be extracted from 25 preselected documents. Overall results on the relevance part of the task were poor, overwhelming the novelty part of the exercise with noisy input.

While our approach is aimed at multidocument summarization, most summarization systems are based on locating similarities between documents [13], [7] and [12]. A group at CMU [6] uses cosine similarity of vectors in the MMR algorithm, which tries to balance relevance to a topic with novelty. Radev[15] uses a similar technique to impose a redundancy penalty in centroid-based summarization. A graph representation of several relationships between words is used to find similarities and differences between pairs of articles [11]. They recognize that sentences cannot be examined independently, without reference to other sentences in the same article.

4 System Description

The system input is a sequence of two or more articles of annotated text on a single event. A parameter indicates where in the stream of input documents it should begin selecting new passages, dividing the input into background and current sets. It can compare a single article against another, or two sets of articles, one set from $day_{m...n}$ and the other from $day_{1...m-1}$.

It expects the input articles to be divided into syntactic units, such as clauses in our experiments, but alternatively in sentences, or into smaller phrasal units. At a minimum, the annotations include part-of-speech tags, and the uninflected roots of the words. We currently use IBM's Talent[16] to locate sentence boundaries, do the part-of-speech tagging and find the uninflected word forms. We also use Talent to identify named-entities, and by running in a batch mode, for the entire cluster concatenated into one file, we get cross-document co-reference. The articles are then run through a finite-state clause recognizer that divides sentences into clause units, each a structure containing a verb and its arguments.

The output is a selection of passages that contain the new information. These can be larger segments than the working units in order to give the user surrounding context, or the units themselves without the context.

The overall idea is to compare what entities of document d_n are covered by documents $d_1...d_{n-1}$. What is not covered is therefore *novel*. The entities can be named or not, abstract or not, and actions or objects – in short anything realized by a content word. In our representation, entities are not only single words, but also groups of equivalent words.

To compare documents, the system compares relationships among the entities. Thus, our representation of the documents must not only group like words together, but also list what entities interact with each other. Then the system can check the entities and their lists of interactions against the earlier documents, the *background*, and determine efficiently which are covered and which are not. Each interaction between entities is essentially a *standin* for a fact, or fine-grained event. The key to this strategy is how to determine *coverage* of such facts.

We seek to make the coverage judgment on the basis of surface information. An interaction or relationship between two entities e_i and e_j is *not covered* if it exists in the current document, d_{curr} but not in the background bg. A relationship is defined as existing between two entities if two words referring to those entities occur in some clause c. We don't try to specify the semantic roles involved in relationship. For example, if "Blake", "Bakley" and "kill" appear in the same unit, we make no effort to specify which is the victim, assuming instead that the inputs will agree on on who killed whom. We just record each possible pair of entities.

$$\begin{aligned}&Nov(e_i, e_j) = True,\\&IfR(e_i, e_j) \in d_{curr}, R(e_i, e_j) \notin bg,\\&WhereR(a, b) \rightarrow a \in Clause_c, b \in Clause_c\end{aligned}$$

Our hypothesis is that we do not need deep language understanding because the system will work with input documents grouped together by a reliable clustering algorithm. Further, we expect only one sense of polysemous words to appear in one set of articles.

The procedure decomposes a document into structures that make it easy to compare to the previous documents. By doing this transformation, we avoid making pairwise similarity judgments of syntactic units, such as sentences or clauses. Instead we can build a structure for all the entities in a document, and with each one, we list all the relationships it has with other entities. Then we can compare these structures efficiently.

The system makes three passes over the input (See Figure 1). The first pass enriches the inputs with semantic information so that individual words that have the potential to point to the same entities are grouped together. The second pass builds up the internal representation of the document, recording the relationships mentioned above. The final pass makes the comparison of the information in the current document to the background of cumulative information in previously seen documents. In the first pass, the system reads the articles sequentially and scans the units in each document, using a semantic database to creates equivalence classes of words that can refer to the same entity, and may use other kinds of semantic or pragmatic information in the future. We call these *Referential Equivalent Classes* or RECs. The RECs are not composed exclusively of synonyms but of all potential referents, including hypernyms, hyponyms and later pronouns. We have experimented with various combinations of resources to help build databases, including WordNet[14], Celex[2], Nomlex[10], and a dictionary built automatically[9]. Here are some example RECs from the pair of articles on the actor Blake:

$$
\begin{aligned}
&REC_i = \{bail, bond, release\} \\
&REC_j = \{condition, lot\} \\
&REC_k = \{dine, eat\}
\end{aligned}
$$

In addition, we create a *REC* for all proper nouns, such as *Robert Blake*, which would include all variations of his name as identified by the named entity recognizer. In the long run, we will create links between the *RECs* for named entities and those for common nouns, so that “Blake” will be recognized as “the actor” or the “star”.

In the second pass, an internal structure is built for each *REC* in each Document representing relationships between *REC*s. We call these *Concept Vectors*, or *CV*, each of which has a *REC* as a head and a list of all *RECs* that co-occur with the head in some clause. We then have a detailed snapshot of what is said about each *REC*. The list of *CVs* implicitly represents all the statements in a document; each *CV* in effect holds a list of facts; each fact is an assertion that the head of the *CV* is in some relationship with all the *RECs* in the *CV's* list. This enables a straightforward method to find which statements in the new document are covered in the background and which are not.

In the third pass, the system performs a comparison of each $CV_{rec,d}$ in the current document d with a cumulative *Background CV*. What we mean by cumulative is that it is updated after each new document is scanned. The program reads through the list of *CVs* in d. If the current *CV* contains an *REC* r not found in the corresponding *Background CV*, a clause containing the head *REC* and *REC* r is considered novel. If there is no corresponding *CV* in the *Background*, then every mention of the *REC* is considered a novel bit of information and added to the output.

```
Background = {}
Novel = {}
1. Read Documents
        For Each Word w
                Fit w in some REC r
2. Reread Docs
        For Each REC r in Doc d
                If no CV[r][d]
                        Create CV[r][d]
                CV[r][d] ← r
3. Reread Docs
        For Each CV v in Doc d
                Test v against Background
                        For Each REC r in v
                        not in Background [v]
                                Create Struct[r][v]
                                Novel ← Struct[r][v]
                Background[v] ← v
Output all Structs in Novel
```

Figure 1: Algorithm for determining novelty.

For example, in the pair of documents about Blake, at one stage in the processing, here is the *Background CV* for "arraignment":

$$CV_{bg}(arraignment) = \left\{ \begin{array}{c} Bakley, Blake \\ accuse, conspiracy, \\ count, courtroom, \\ murder, plea, shoot, \\ guilty, solicit \end{array} \right\}$$

and as the next article is processed, its *CV* for "arraignment" included the novel subset:

$$CV_{curr}(arraignment) \supset \left\{ \begin{array}{c} Caldwell, trial \\ former, innocent \end{array} \right\}$$

Because the co-occurrence of Caldwell and arraignment appears in the new article but not in the background article, this clause is retrieved as new:

> Also at Thursday arraignment hearing, Blake former handyman-bodyguard, Earle Caldwell, pleaded innocent to a murder conspiracy charge

5 Data Annotation

Before we could begin to refine our ideas about the problem, we needed a sufficient corpus of unbiased novelty decisions. This proved to be an expensive proposition. It

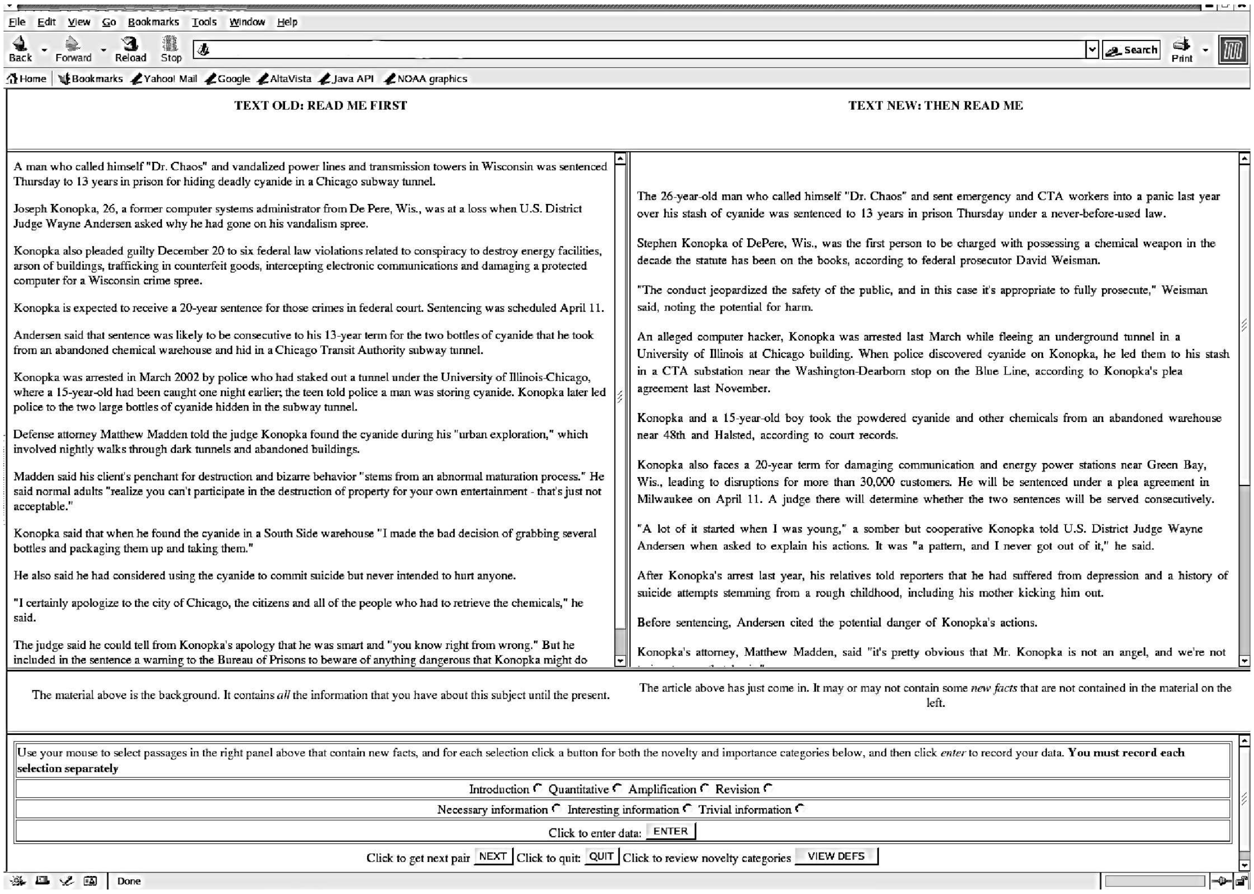

Figure 2: Example of the annotation interface, displaying a pair of articles and the form for the annotators

was essential to eliminate, as far as possible, disagreements between annotators. In a pilot annotation, done some months earlier, we made many mistakes to guide us in constructing a better testbed, a collection of pairs of news articles.

Each pair of articles was on a particular event; some examples are the announcement of Marlon Brando's divorce settlement, the marriage of an inmate on death row, the lawsuits filed by America On Line against spammers, and the proceedings against Robert Blake. We aimed to find articles that largely covered the same ground but differed in detail. The articles were usually published in short spans of time, often within a day of one another, and often published by different news outlets. We favored articles about more obscure events to force the annotators to rely on the texts rather than to draw on their own knowledge of the events of the day.

Our pilot annotation was done by volunteers. It was marred by low agreement and varying degrees of effort on the part of the volunteers. This time we asked paid annotators to make three passes over the material. In the first pass, they used an *HTML* interface available over the *World Wide Web* (See Figure 2). To keep the task simple, we displayed the pairs of articles side by side, and we had the annotators use the mouse to select fragments of text that contained new information. We did not define any particular structure to be marked – like sentences or clauses. We recorded hidden indices adjacent to each word. For each selection, the annotators were forced to grade the selection in two ways: *novelty* and *importance*. We gave them these choices to respond to the two questions.

Type of Novelty Introduction, Quantitative, Amplification, Revision

Degree of Importance Necessary information, Interesting information, Trivial information

By separating the two judgments we avoided clouding the question of novelty from the more subjective notion of importance.

After the first pass, we automatically created new files, coloring the areas where the two annotators agreed and where they disagreed. Then, they were asked to reread the articles and reconsider all the areas marked as disagreement. On one set of five pairs of articles, our annotators disagreed on 48 fragments, almost half of the fragments initially marked. But in this second reconsideration pass, 35 of those were resolved – simply by having the annotators take a second look, with the disagreements highlighted – leaving only 13 points of disagreement.To resolve these remaining points, they began an exchange of e-mails; in this case 9 points were resolved in the first exchange, and the last 4 in the final exchange.

Here is one passage from the Blake article, where one annotator marked the bold face section, and the other didn't on the first reading. The two remained in disagreement after the reconsideration.

> **Appearing in court for the first time since he was allowed to post bail March 14**, a healthier looking Robert Blake pleaded innocent to murdering his wife and waived his right to a speedy trial until October, when proceedings are likely to begin.

But the annotator who favored novelty, argued in the negotiation phase, "The 1st article didn't say this was the first time he had appeared in court since posting bail." With that, she persuasived her counterpart.

At the end, it became clear that *novelty* was not a terribly subjective quality, but that the markup work was subject to lapses on the part of the annotators. These were easily corrected by having them review their work.

6 Experiments

We conducted experiments to explore the effect of using different types of syntactic analysis and of using variations of our semantic database. We also compared our overall approach to a simple sentence-based system. We tested the different system variations on all 31 pairs of articles that had been annotated in a first round in the early summer. The system was constructed without reference to these test articles, and we are in the process of conducting a new round of annotation and we will hold those pairs out for future tests on changes to the system. In these experiments, we ignored the annotators' assessments of importance. We did this for two reasons. First, metrics of importance and novelty can work against each other. Second, our goal at this time is to isolate the indicators of novelty. Later on, we will use techniques in multi-document summarization to measure the relative importance of novel passages.

In addition, we combined the results of our system, which analyzes documents that have been broken down into clauses, and the sentence-based system, which does a pairwise comparison of sentences, and achieved a substantial jump in precision, to 0.73 (Table 1). This is well ahead of a random system, which would be expected to obtain a 0.63 precision, and better than either strategy alone attained (Tables 2 and 3). This result is in line with an observation we made about the data, i.e. the annotators selections. Sometimes they chose a single clause or phrase as the new material. Any sentence based strategy would have difficulty because the two parts of the sentences then work against each other. But other times the annotators often chose novel segments that spanned several sentences, giving an advantage to the sentence-based system, which would pick up on the new terminology that accompanies a topical shift.

Threshold	Prec	Rec	Size
Cos=0.10	0.73	0.22	361
Cos=0.20	0.71	0.32	549
Cos=0.30	0.69	0.40	701
Cos=0.40	0.67	0.42	766

Table 1: Results of taking the intersection the vector-space model and our system with the WordNet plus Celex database, and manual pronomial resolution.

Thus it appears that our system helps the vector-space program by eliminating parts of sentences that were not judged as novel. To illustrate the effect, consider the sentence from the Blake article below. The numbers in parentheses show the clause boundaries found by our parser. The novel material, according to the annotators, was the fact that this was his *first* appearance since posting bail, which is contained in the first three units. In this segmentation, we can also pinpoint that it was Blake who waived his rights to a speedy trial and not his wife. The sentence-based system selects this sentences when the similarity threshold is at 0.3 and higher. At that level, the sentence-based system would return 89% of the original article, when only 69% was judged new by the annotators.

> **(1)**Appearing in court for the first time **(2)**since he was allowed **(3)**to post bail March 14, a healthier looking **(4)** Robert Blake pleaded innocent to murdering his wife **(5)** and waived his right to a speedy trial until October, **(6)** when proceedings are likely **(7)** to begin.

In turn the vector-space helps our system by focusing on the sections that are richer in new material. In its present form, our system is quite susceptible to noise. Since it makes binary decisions on all combinations of words, accepting whenever there are two words together that are not found together in the background, there are many opportunities for false positives.

The example also shows the need to carry forward some words, like heads of the subject noun phrase, to the next clause, like those headed by nonfinite verbs or in cases of ellipses, and that we need pronomial resolution.

6.1 Semantics

We tested several alternate semantic databases to gain some insight into how to increase coverage without adding noise. WordNet is an obvious place to start, but writers do not limit themselves to synonymy in choosing referring expressions, for example "dog" and "pet". This seemingly common association cannot be retrieved from WordNet. Consider, the immediate hypernyms of the six senses of dog in WordNet2: "canine", "unpleasant woman", "chap", "villian", "catch", "support". It quickly becomes clear that we have to expand its reach while removing obscure links. Our semantic dictionary is used by the system to build the *RECs*. The results on the choice of database are listed in the first column of Table 3.

The five alternatives are:

DekLin The database consists exclusively of Dekang Lin's[9] thesaurus of similar words. He clusters words automatically on the basis of their distributional patterns.

Empty A database that only identifies the uninflected roots of each word.

BaseWN A database composed entirely of WordNet [14] synonyms, hypernyms and hyponyms.

WN+DL The intersection of extended links in WordNet *and* DekLin's thesaurus.

WN+CLX WordNet plus derivations, such as nominalizations of verbs from Nomlex [10] and CELEX[2].

Although the detailed results show numerous differences over the test pairs of articles, they tend to even out. There was no significant net gain in adding any of the extra information over the Empty database. Only the automatically built dictionary contains enough noise to degrade performance somewhat. In fact, all of the databases contain considerable noise because they contain infrequent and even obscure links between words.

Agreement on the clauses was obviously quite high, and we computed the Kappa co-efficient at 0.75, but there were sufficient differences to support much greater variation in the scores. For example, there were 568 differences among the system. 425 were 5-to-1 splits, 92 were 4-to-2 splits, and 51 were 3-to-3 splits. It seems that each variation simply adds some more information and more noise at the same time.

We also resolved pronomial references manually into a copy of the data. We had anticipated that we would ultimately need a module to resolve pronomial reference and to link named entities to noun phrases, but have not yet written that module. The second

Threshold	Prec	Rec	Size
Cos=0.05	0.62	0.35	675
Cos=0.10	0.65	0.47	866
Cos=0.20	0.66	0.77	1405
Cos=0.30	0.65	0.93	1722
Cos=0.40	0.64	0.98	1863

Table 2: A simple vector-space model applied to whole sentences, and computed with the cosine metric.

Database	NIA Only		NIA+Prons		NIA+Charn		NIA+Pro+Ch		Summary Size
	P	R	P	R	P	R	P	R	
DekLin	0.57	0.25	0.63	0.30	0.62	0.25	0.63	0.29	586
Empty	0.61	0.36	0.64	0.42	0.64	0.37	0.64	0.39	785
BaseWN	0.62	0.38	0.64	0.43	0.64	0.38	0.65	0.41	803
WN+DL	0.60	0.37	0.64	0.43	0.63	0.37	0.64	0.40	814
WN+CLX	0.61	0.37	0.64	0.43	0.64	0.39	0.65	0.41	804

Table 3: Results our system in various configurations.

column in Table 3 shows the result for the five databases after pronomial resolution is added; both precision and recall increase. The differences in precision between the databases all but disappear and only DekLin lags in recall. This result raises the immediate question of how much of gain this will translate to when we include an automatic system of pronomial resolution. There are a number of systems available, and it seems that they function at about 70% accuracy. In the combination experiment, we used the version of our system with pronomial resolution, and with the WordNet plus Celex database.

6.2 Parsing

We also tested a more powerful syntactic analyzer. The system was intended to use a finite state clause recognizer that we use in some summarization tasks and in corpus analyses. It has the advantage of being fast and reasonably accurate, but we wanted to measure the gain by using a more powerful tool and tried the probabilistic parser from Brown University [3] without and with the manual pronoun resolution.

With the full-scale parser by itself, Column 3 of Table 3 shows a boost in performance, particularly in precision over the basic system. But while equal in precision to the pronoun-resolution version, it is somewhat behind in recall. When we combined the more powerful parser with pronoun resolution, we saw progress inch foward again, as seen in Column 4 of Table 3. Two factors limit the potential gain from this combination: First, we are still dealing with inadequate semantic resources, and second, we are not yet filling in gapped or elliptical references.

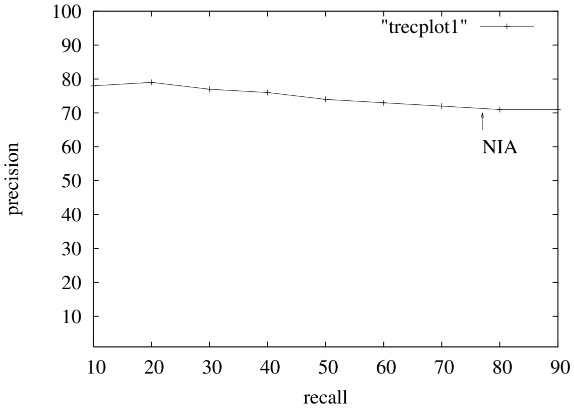

Figure 3: Trec Novelty Track data shows little precision/recall tradeoff

6.3 Sentences

Our sentence-based system is the traditional vector-space model that uses cosine distance to measure similarity, and we considered novelty to the inverse of similarity: $1 - Cos(s)$. The program ignored a list of stop words and imposed no weighting of the content words, such as $tf * idf$, and did quite well in comparison to our system. Table 2. The curious aspect of this program was that precision never changes. It actually sinks at a very high threshold for novelty. Recall behaves in the expected way, coming close to 1.0 at a middle-level threshold. Note that precision and recall at the threshold of $Cos = .10$ is very near the performance of our system, and produces the same sized summary. Each clause in a selected sentence was counted for scoring purposes.

In addition we ran the TREC 2003 data through our vector-space program to see if we would also get a flat precision curve. The TREC 2003 data contained a much larger portion of novel sentences in the list of relevant sentences, 10,226 out of 15,557, or 65.7% than in TREC 2002. However, a substantial number of the non-novel sentences are exact duplicates, as there are a fair number of duplicate articles in the topics. Figure 3 shows a very flat precision curve hovering slightly above random (after factoring out the duplicate articles). We ran our system on the data, but as might be expected, did poorly being forced to select full sentences listed out of context.

7 Conclusion and Future Work

We have worked in our experiments to explore the dimensions of a new problem in computational linguistics. We sought to show that finding new information would require a greater amount of syntactic and semantic analysis, and, indeed, we achieved our best results when we combined a standard model of comparing sentence vectors with our more fine-grained approach.

It seems that new information comes in different granularities. There are details entirely contained in a small phrase and statements entirely contained in clauses, which can be embedded within a sentence or can comprise a whole sentence. Then there are larger segments, where some novel subtopic is introduced. Thus we achieve better performance with a mixed system rather than one focused on a single level of granularity.

We learned that we must go much farther in reference resolution. We have shown that we need to link pronouns to the terms they represent. In addition, it seems clear that we must link named entities and the common nouns that refer to them. The challenge will be to find an automatic means of performing reference resolution that, given current accuracy levels, does not add more noise than information.

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] CELEX. *The CELEX lexical database — Dutch, English, German*. Centre for Lexical Information, Max Planck Institute for Psycholinguistics, Nijmegen, 1995.

[3] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the NAACL-2000*, 2000.

[4] 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.

[5] D. Eichmann and P. Srinivasan. Novel results and some answers: The university of iowa trec 11 results. In *Proceedings of the 11th Text Retrieval Conference*, 2002.

[6] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL-2000 Workshop on Automatic Summarization*, 2000.

[7] S. Harabagiu, D. Moldovan, P. Morarescu, F. Lacatusu, R. Mihalcea, V. Rus, and R. Girju. Gistexter: A system for summarizing text documents. In *Proceedings of the Document Understanding Conference (DUC01)*, 2001.

[8] K. Kwok, P. Deng, N. Dinstl, and M. Chan. Trec 2002 web, novelty and filtering track experiments using pircs. In *Proceedings of the 11th Text Retrieval Conference*, 2002.

[9] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, 1998.

[10] C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. reeves. Nomlex: A lexicon of nominalizations. In *Proceedings of EURALEX'98*, 1998.

[11] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. In *Proceedings, American Association for Artificial Intelligence 1997*, 1997.

[12] D. Marcu. Discourse-based summarization in duc-2001. In *Proceedings of the Document Understanding Conference (DUC01)*, 2001.

[13] K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformation: Progress and prospects. In *Proceedings of American Association for Artificial Intelligence 1999*, 1999.

[14] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, 3(4):235–312, 1990.

[15] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation and user studies. In *Proceedings of ANLP/NAACL Summarization Workshop*, 2000.

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