Experiments in Novelty Detection at Columbia University

Barry Schiffman
Department of Computer Science
Columbia University
bschiff@cs.columbia.edu

Abstract

This paper describes the method we used for the Novelty Track for the 2002 Text Retrieval Conference (TREC). We tried to adapt tools we are developing for a task closely related to the novelty part of the this track. The system we are building will scan a stream of documents and present to the user only the new information it finds. For the "relevance" part of the TREC, we decided to test the applicability of some of these tools. Since information retrieval is not a focus of our research, we thought it would be more interesting to use something new rather than try to hurriedly catch up. The results were far from satisfactory, but it is clear from the overall results that novelty detection remains a difficult and unsolved problem.

1 Introduction

The task in the Novelty Track at the 2002 Text Retrieval Conference (TREC) was structured in two parts. First, the system had to find sentences in a cluster of documents that are relevant to a query, and second, as the sentences were presented in a predetermined order, it had to remove any that duplicated information in previous sentences. The clusters themselves were culled from the fourth and fifth TREC collections by an Information Retrieval system, selecting the documents relevant to the query. The queries were 50 previous TREC topics, in some cases altered somewhat. Up to 25 documents were collected for each topic.

Our interest in participating in the Novelty Track was to work with the data in the second part. We are building a system, called the New Information Agent (NIA) to detect new information from a stream of document. Like the TREC version, the input to our system is a clustered stream of documents, or in an offline version, a collection of documents, that focuses on a particular event or issue. Again, like the TREC task, the output of our system is a short list of the sentences that do not contain any material that duplicates a passage selected earlier. But the presence of the query is the key difference between the TREC version of the task.

We consider all the documents to be of potential interest to the user. Because the query dominates each problem, the TREC task calls for deciding relevance first and novelty second – the reverse of what we will do in our system. We first identify segments that contain new information and then decide if they are interesting. In our terms, interesting is not the same as relevant, since we have no query to base relevance on.

With a query, the task is more focused, providing the system with some kind of guide for what to select, but the characteristics of the tasks vary with the kind of topic used. The sample topics suggested that deep understanding of language would help, and might even be necessary for strong performance. For example,the first sample, about the Hubble Space Telescope, asked for material about the achievements of the telescope and not material about repairs or modifications to the telescope. We know of no automated system that can classify events as achievements or not achievements in relation to an arbitrary object, here a telescope. It seemed clear that the relevance portion would dominate the task. The coordinators of the novelty track said so when the guidelines were promulgated.

Because we have no experience with relevance judgments, we chose to experiment with an unusual approach that borrowed the language analysis tools we are developing for our new information system.

1.1 New Information

NIA analyzes a document in terms of the content words and the contexts in which each one appears, and then compares documents by comparing these contexts in structure called *Concept Vectors*. In order to build these *Concept Vectors*, the system groups the words into sets of "referential equivalents" or *Concept Sets*, so that in a document about the Hubble space telescope, the words *telescope* and *instrument* would be equated and put into the same *Concept Set*. The *Concept Vectors* are creating by making lists of which *Concept Sets* co-occur with each other. These vectors are compared across documents – not sentences or clauses.

The system uses a syntactic analyzer that breaks up documents into clause-sized chunks. These are used in two different ways: 1.) potential "equivalents" are grouped together only if they appear within n clause chunks of one another, and 2.) segments of new information are identified by examining the concepts in each clause with respect to how well their corresponding vectors are covered by previously seen material. In our version of the new information task, we hypothesize that sentences are not a good unit for analysis. Rather than consider the similarity or dissimilarity of whole sentences, we are trying to efficiently decompose the documents into small chunks and discover when new relationships between entities appear.

In the TREC task, we lost that framework since the novelty part examines a collection of sentences that relate to a query, but are each individual passages taken out of context. The result is that the system we are developing is not appropriate and was ignored. In addition, we were running out of time, so that the novelty part of our task was done with a rather simple system of computing the overlap of the words sentence by sentence.

In the rest of this paper, Section 2 will discuss work related to our experiments; Section 3 will talk briefly about the system we are building; Section 4 will provide a desciption of the program used in the Novelty Track; Section 5 will review its performance; Section 7 will reflect on the lessons learned.

2 Related Work

Novelty detection is a new area of research, with roots in information retrieval, in particular first story detection under the Topic Detection and Tracking (TDT) initiative and in multi-document summarization. The task defined in the TREC Novelty Track is closer to the TDT task. Some recent work by James Allan exemplifies the extension of TDT to the passage level of documents (2001). He posit that a sentence is "useful" if it is on topic, and that a sentence is "novel" if it is not redundant with previously seen sentences. Their perspective is topic-based and the experimental corpus comes from the TDT-2 corpus, in which 60,000 news stories were assigned to some 200 news topics. After selecting 22 of these topics, annotators created lists of the events that comprised each topic and assigned each sentence to one or another event. A total of 343 events were derived from 944 articles. Two different language models for deriving "useful" information were developed, based on the probabilities that individual words of a sentence appear in on-topic sentences or articles. The models of novelty are derived in a similar way from the specific words in on-event sentences.

A numer of efforts in multi-document summarization have sought either to highlight differences or avoid redundancy. A group at CMU (Goldstein et al., 2000) uses cosine similarity of vectors in the MMR algorithm, which is cited by Allan. They seek to eliminate redundancy from their summaries with a measure similar to Allan's novelty detector. Radev attempted to create a framework for analyzing differences between sentences between sentences from different documents, with relationships such as "equivalence", "subsumption" or "contradiction" (2000).

A graph representation of several relationships between words is used to find similarities and differences between pairs of articles (Mani and Bloedorn, 1997). They recognize that sentences cannot be examined independently, without reference to other sentences in the same article. A group from Cornell and Cogentex is looking at the related problem of "discrepancy detection," in particular those of numerical differences (White et al., 2001).

The structure of the task in the Novelty Track is close to the work of Allen and that of Goldstein, although

they had used a linear combination of both relevance and novelty qualities, but it requires separate computations. Our developing work views a document in a way close to Mani and Bloedorn, but unfortunately it could not be directly applied to this task.

3 Overview

The query-based structure of the Novelty Track prohibited the direct use of our system, NIA. Queries contain only general statements about the topics, and a perfectly functioning NIA would return all the details in the set of documents as *new*. But we wondered if we could apply the *Concept Sets* and the syntactic analyzer to both the relevance and novelty parts of the Novelty Track. This strategy was problematic since NIA has no machinery to determine relevance to a given query. NIA is intended to track a topic or event over time and provide updates. It assumes 1.) that the input documents are clustered appropriately, and 2.) that the user cannot predetermine what aspects of the topic or event will be interesting. But, on the other hand, the exercise might offer much insight into the performance of the tools we are developing and might ultimately be more beneficial to us than trying to quickly patch together an information retrieval system.

The borrowed tools include the lexicon used to build the *Concept Sets*. It provides what we call "potential referential equivalents", that is words that can be used to refer to one another. In addition to it, we compiled a lexicon of associated words drawn from a background corpus of news and combined these elements in a rule-based system that made a relevant/not relevant decision on each sentence in the document cluster.

3.1 Sample Sets

Like the other participants, we had only four sample sets for development, and used those to design and tune the system. The prospects were challenging. It was obvious that the four samples were quite diverse. Further, it was difficult to guess about the test data since the track organizers intended to alter the wording of some of the topics in the actual test and since we had no idea which documents might be listed as the most relevant.

We also noticed in the sample sets that the annotators' tended to favor a few of the documents. Based on that observation, we built the system to automatically decide if a few documents strongly addressed the issue in the topic. Where that was the case, we drew all our relevant sentences from those central documents.

Finally, we developed the parameters our system uses by experimenting on the sample sets. We sought to balance the recall and precision on the sample sets, and we aimed to present summaries of reasonable size, given the examples, and avoided submitting either very small or very large summaries.

The sample sets themselves were interesting. Here are some observations we made from an initial look at the problem:

Hubble Strong performance here seemed to depend on a clear idea of what is and is not an accomplishment. There were some useful key words, like data and theories, but the set contained a number of off-topic articles that were not likely to discuss Hubble's accomplish, including those on a species of squirrel and on a big earth-bound telescope being built by the Europeans.

mutual funds The system needs to know what a predictor is. There is a conflict in the language. In the *description* it says "predictors of mutual fund performance (excluding issues of costs and yields)" and in the *narrative* it says "a documnet must contain at least one factor such as: rankings, risks, yields or costs". Our initial tests were not able to suggest a strategy for this set, but it was described as atypical.

mainstreaming The interesting aspect here was that the word *mainstreaming* rarely occurred in the document set ($< 1\%$ of the sentences), but only 3 times in the relevant sentences, forcing the system to rely on the terms "children", "impairments" as well as to have an understanding of "pro" and "cons".

Mirjana Milosevic Strong performance here was attainable simply by scanning for sentences that mention the woman's first name, or nickname Mira, . Other strategies diminished these results.

4 System Features

4.1 Relevance

Most of our system-building effort went into the relevance part of the task. We settled on a rule-based approach, rather than a vector-space approach. We expected that most participants would be far more experienced in information retrieval methods and would be in a much better position to refine them to this task. Thus we viewed our submission as an opportunity to test unusual ideas that were more closely related to the thrust of our research. Admittedly, this gives our system a patchwork quality, but one that would hopeful provide valuable insight into alternative approaches.

A number of features are computed for each sentence, and sentences are selected if the rule is satisfied. We submitted five runs, using different combinations of rules and parameters. Development of this system was based almost entirely on four samples.

1. distance from a title word in a prominent role in a clause (target distance)
2. word match with a potential referential equivalent (equivalent count)
3. word match an associated word (associated count)

The first feature is binary, reflecting whether the current passage is near enough to the previous *prominent* mention in the document of a term that appears in the title. Passages were either clauses or sentences, and promiment means that the target word appears as a standalone NP before the verb.

The second and third features refer to the two lexicons mentioned above. The values are just raw counts. We observed that the clause chunks are uniformly short and that the appearence of both "equivalent" words and "associated" words is relatively rare.

We tried a number of other features, and ended up ignoring several. The ones retained were based on the various fields in the topics, such as "titles", "narratives". The three used are:

The lexicon that provides potential referential equivalents is a new version of the of a resource we have been using in NIA. There, it is used to build *Concept Sets*, which are are words linked semantically. In order to avoid the need to disambiguate among word senses, highly polysemous words are filtered out, and a distance constraint is imposed before words are grouped together in a *Concept Set*. Thus, as the text is scanned, the system checks to see if it belong to an existing set or if it will instantiate a new set. The function to accept a word for inclusion in the Relevance part is:

$$Accept(w_i) = \begin{cases} true & if \ \ senses(w_i) < m, \\ & \ \ dist(w_i, w_j) < n \\ false & otherwise \end{cases}$$

where *senses* is the count of WordNet senses, and *dist* is the number of clauses between w_i and w_j the previous occurrence of a word in the same equivalence class.

Because the Novelty Track task required us to relate the words in a query to those in a document, we were unsure of how to modify technique, since the queries, that is the topics, are too short to allow the building of *Concept Sets*. In the end, we risked injecting noise into the decision-making and ignored the second condition for acceptance. We went forward with this strategy because it seemed to work reasonably well in tests conducted on the sample sets.

The raw equivalence lexicon is built mostly from WordNet (Miller et al., 1990), using synsets, hypernyms and hyponyms. NIA uses nouns and verbs, but we included adjectives for this effort. In the future, the lexicon will be altered with the results of corpus statistics that we are in the process of gathering. It is not clear yet whether we will keep the adjectives.

The lexicon of associated words is based on co-occurrence patterns in a background corpus. The corpus we used was from Reuters in 1996 and might have added some noise to our submission. Using the underlying TREC collections used in the track might have been more effective here, but we wanted to test using an orthogonal corpus, since in NIA we will have no knowledge of future changes in the discussion of a particular topic or event. We also used a clause-level co-occurrence standard rather than a document-level standard, since the task examines and makes decisions on short passages – sentences, which are usually composed of one, two or three clauses. We used mutual information to measure the degree of relatedness between two words.

$$MI(x,y) = log\frac{(p(x,y)}{p(x)p(y)}$$

We also added an adaptive capability to our system, given the different types of topics in the Novelty Track. These automatically assess two characteristics of the document set and adjust the system's behavior to these. In previous research in multi-document summarization, we used a similiar technique in the DEMS summarizer, Dissimilarity Engine for Multi-Document summarization (Schiffman et al., 2002), to good effect in the Document Understanding Conference 2002.

One adaptative method controls the value of the feature that check the target distance feature, the distance between the current passage and the last mention of a word in the topic title. We observed that topics did not always contain usable title words – like mainstreaming – so that the target distance would be self-defeating. For this, we measured the likelihood of finding any target word in the document set. If the total was below a threshold, we set the distance at such a large number that it no longer carried any weight.

The other adaptive method controls the number of documents that were examined. We noticed in the sample sets that in some cases a few documents dominated the selection of relevant sentences, suggesting that cluster contained some documents that were only tangentially related to the topic. In order to discern when this occurred, we used the lexicon of associated words. We computed the likelihood of finding words from any field in the topic and then computed the variance of these likelihoods across the documents in the set. If the "associated-words variance" was below a threshold, we concluded that most of them would contribute to the output of relevant sentences. Otherwise, we concluded that the document set contained some outliers that would be best to ignore.

One final strategy we adopted was to remove words that frequently appeared in a large number of topics – words like "relevant". To avoid having sentences accepted on the basis of these, we computed an inverse topic frequency value for all words in the 150 topics from which the test set would be drawn. These words were eliminated from the topics before the topics were expanded to include the referential equivalents and the associated words.

4.2 Novelty

Unfortunately the relevance part of the task took most of the time we had alloted, and with a limited time left, we adopted a very simple duplication test. From the sample topics, there was little for a novelty detector to do. We first expand each sentence by adding the referential equivalents. We did this despite the risk of eliminating novel sentences because of the appearance of unrelated senses of polysynonymous words. As we considered new sentences, we computed how well the new sentence was covered by each previous sentence and rejected those that exceeded a threshold. The mechanism we developed for NIA would have required us to reference the original documents in order to examine the context of each sentence.

5 Overfitting

We submitted all five runs that we were allowed. All used the same structure outlined in Section 4, but with different parameters. Three of them were based on clauses, that is the features were computed on the basis of the clauses recognized by our clause-tagging tool. We used these to test whether the on-line adjustments had value. The runs marked *cl35* and *cl85* in Table 1 did not try to adapt to the document set. The numbers 35 and 85 refer to the percentage of documents from which the relevant selections were drawn from. The documents are ordered according to their distributions of associated words. The run marked *clfx* automatically selected either the .35 or .85 figure according the variance of the likelihood of finding an associated word in the documents.

The *sent* run computed the features over sentences, and the *merg* concatenated the sentence following any sentences that scored high, to test the possibility that segmenting documents might be a valuable idea. Both of these used the automative adaptation mechanism.

It appears that we were lulled by a painful instance of overfitting. The devlopment of our system was closely guided by its performance on three of the sample sets. The first, topic 303, was described as typical of the entire test. Table 2 shows how the system performed on

	Relevant			New		
	P	R	P*R	P	R	P*R
cl35	.07	.04	.006	.07	.04	.005
cl85	.09	.07	.009	.08	.05	.007
clfx	.07	.12	.012	.07	.09	.009
sent	.11	.09	.012	.12	.09	.012
merg	.11	.15	.020	.09	.10	.013
humans			.191			.170
random			.006			.004
best system			$<$ .095			$<$ 0.85

Table 1: Precision and Recall of our five runs, humans and random

the relevance part. The results seemed to be sufficient on this difficult problem. We didn't think we had a top system, but were satisfied with what we saw.

We didn't include topic 359 for several reasons. In our early experiments, it seemed to be impossible to match any of the human selection and futher more it contained a contradiction: Description 2 wants to exclude costs and yields, but the Narrative wants to include them.

Our results were disappointing even though we did not expect much at the outset. The organizers of the Novelty Track reported that human annotators tested against each other had achieved a score of 0.19 – this is the product of the standard measures of precision and recall – $Score = Prec * Recall$. They said that the best submission was less than half that of the humans, but Table 1 clearly shows all of our runs were far below that.

If an oracle program were able to choose the best system for each topic, the combined score averaged together would be 0.134 on the relevant part and 0.120 on the novelty part. Since this score was a good deal better than the best system, no one system was consistently at the very top.

The topic sets also varied widely, and some were difficult for all systems, other much easier. Averaging the scores by all systems for each topics showed a wide range, indicating some sets were managable for a number of systems, while others were nearly impossible for all of them. Assuming that the average of all systems indicates the degree of difficult we have:

On average, the novelty task proved to be much harder. There was a striking drop off in the average scores. This is surprising since the annotators eliminated very few relevant sentences in creating their list of new sentences. In fact, a baseline that does nothing – that does not eliminate any relevant sentence – would have a precision of $.91$ and a recall of $.99$. (The recall appears to be short of 1.00 because the relevant and new lists were swapped in two cases.)

Just before the paper submission deadline, the Novelty Track organizers restated the results, using the standard F-measure instead of the product of precision and recall.

$$F = \frac{2PR}{P+R}$$

The recalculation raised the single-value scores of all groups, and squashed the results into a much narrower range for the automatic systems, as Table 5 shows. The recalcuation also tended to eliminate the size of the advantage to systems that generated larger summaries.

6 Recent Experiments

To explore performance in the novelty part further, we counted duplicates found, rather than novel sentences found. In the formal task, the scores in the first part address the question of "How many of the relevant sentences can the system find?" The scores for the second part address the similar question of "How many relevant (and nonduplicative) sentences can the system find?" The question makes more senses in the relevance part where only a small portion of the sentences are judged relevant. In the four sample top-

Results on Sample Set									
	Clause-based *clfx* run			Single sentence *sent* run			Paired sentences *merg* run		
Topic	Prec	Recall	P*R	Prec	Recall	P*R	Prec	Recall	P*R
303	.636	.438	.279	.667	.250	.167	.261	.375	.098
379	.194	.235	.046	.250	.216	.054	.238	.294	.070
423	.211	.160	.034	.786	.147	.116	.545	.320	.174
	average $p * r = .120$			average $p * r = .112$			average $p * r = .114$		

Table 2: Precision and Recall achieved by our system across the three sample topics for detection of relevance.

Relevance			
Easiest Topics		Hardest Topics	
Topic	Score	Topic	Score
368	0.262	312	0.019
397	0.247	381	0.018
394	0.193	305	0.018
365	0.189	432	0.017
369	0.167	420	0.016

Table 3: Average $P * R$ scores on the Relevance part show a wide range of difficulty.

Novelty			
Easiest Topics		Hardest Topics	
368	0.127	445	0.005
397	0.108	312	0.005
394	0.103	432	0.003
365	0.089	377	0.002
364	0.080	420	0.0002

Table 4: Five highest average $P * R$ scores in the Novelty Part of the task.

F-Measures	
Summary generator	F-measure relevant
Humans	0.371
Top Sys	0.235
Best Novcol	0.126
Random	0.040

Table 5: Restatement of some results on the Relevant part of the task in terms of the standard F-measure. The 10 top scores plus human and random results were distributed by NIST before the paper-submissions were due. The Novcol were recomputed.

ics given to participants, about 6% of the sentences were accepted as relevant. The situation was the reverse for the novelty side, where nearly all the relevant sentences were considered novel. In the test, the annotators removed only 106 of 1,347 sentences were removed as duplicative. With a lopsided test set, it is hard to beat the baseline of "do nothing." So we recast the question into "How many duplicative sentences can the system find?" We then computed precision and recall for a number of baselines, including a bag of words approach, TF*IDF, longest common subsequence, and Simfinder, another tool for measuring similarity between sentences developed at Columbia(Hatzivassiloglou et al., 2001). Table 6 shows that our semantic module outperformed the other methods. We show the results when the parameters for the various methods made a reasonable number of selections. By making the duplication thresholds low enough, most of these methods will choose a large proportion of sentences as duplicates and achieve a high recall.

Here are descriptions of the methods presented in Table 6..

novcol Our semantic module applied to whole sentences.

sequent The longest common subsequence of words, as a percentage of sentence length.

wordbag A unweighted bag of words approach, using overlap.

similar The Simfinder utility..

tfidf TF*IDF metric [1] and computing cosine similarity. Document frequency values were taken from the set of articles for the topics.

random An extrapolation of random results by computed the expected value of 106 selections.

7 Conclusion and Future Work

One positive lesson learned in this exercise is that the adaptive strategy appears to have considerable value. Without sufficient training data, it was impossible to explore and sharpen the technique, yet it clearly improved our results in the runs where it was applied, despite having only a rough idea of the parameters to use. In addition the range of averages across the topics suggests that a one-size-fits-all approach is not the best.

Our experiments after the evaluation show there is a value using semantic information in detecting similarity and dissimilarity. This was not so clear about our application in the relevance part of semantic data – in the form of the lexicon of referential equivalents. We were hampered because our system was unable to apply the lexicon in the way it is used in our NIA system, where the expansion of the sets is limited by the context in the documents to be summarized. Since the topics were too small to provide any context, the lexicon was used without distance constraints. But in the more straightforward task of detecting duplication, the semantic information without those constraints

An assessment of using associated words – those obtained by co-occurrence studies – was clouded by the fact that the data was drawn from a much different collection of background documents. This was due to a lack of time. We had a collection of Reuters news wire already parsed, and would have had to delay experimentation if we had waited to parse the TREC collections used in the Novelty Track. We are planning to create a new lexicon based on the TREC documents to compare against the results here.

The Novelty Track also confirmed how difficult the task is. The subjectivity of the annotation greatly complicates the conclusions that can be drawn. Judging from the cross annotator scores, inter-annotator agreement was quite low, and the choice of annotator may have had a large effect on the results in various sets.

We were also struck by the fact that many, but certainly not all, topics included some instruction about was not relevant. We blocked those that were found in such negative sentences from being expanded if they were not already found in the positive sentences – however these were few in number in the sample sets. We did test a feature of noting the presence of negative terms in the passages, but where it did affect the outcome, it was detrimental as often as helpful. Yet, we think the idea of trying to categorize the queries, that is the topics, is worth further experimentation.

[1] $weight = (1 + log(tf))log(idf)$

	Matched	Sys Tries	Hum Picks	Prec	Recall	P*R
novcol	34	139	106	0.2446	0.3208	0.0785
sequent	30	156	106	0.1923	0.2830	0.0544
wordbag	24	107	106	0.2243	0.2264	0.0508
similar	25	158	106	0.1582	0.2348	0.0373
tfidf	14	126	106	0.1111	0.1321	0.0147
random	8.3	106	106	0.0783	0.0783	0.0061
do-nothing	0	0	106	0	0	0

Table 6: A comparison of results on duplication detection between the semantic module and several word-based methods and a system that would choose at random. Note "do-nothing" gets zero because it selects no duplicates to reject.

References

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of news topics. In *Proceedings of the ACM-SIGIR Conference*.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL-2000 Workshop on Automatic Summarization*.

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. 2001. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the NAACL 2001 Workshop on Automatic Summarization*.

Inderjeet Mani and Eric Bloedorn. 1997. Multi-document summarization by graph search and matching. In *Proceedings, American Association for Artificial Intelligence 1997*.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, 3(4):235–312.

Dragomir Radev. 2000. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *1st ACL SIGDIAL Workshop on Discourse and Dialogue*.

Barry Schiffman, Ani Nenkova, and Kathleen McKeown. 2002. Experiments in multidocument summarization. In *Proceedings of the Human Language Technology Conference*.

Michael White, Claire Cardie, Vincent Ng, Krii Wagstaff, and Daryl McCullough. 2001. Detecting discrepancies and improving intelligibility: Two preliminary evaluations of riptides. In *Proceedings of the Document Understanding Conference (DUC01)*.