Towards An Adaptive Framework for Information Retrieval

Scott A. Weiss
Simon Kasif
Eric Brill
{weiss, kasif, brill}@cs.jhu.edu
Dept. of Computer Science
The Johns Hopkins University

October 31, 1995

Abstract

We report on our investigation into techniques for adaptive information retrieval. We describe our domain of USENET newsgroups, and discuss some of our inital experiments. We illustrate the weakness of naive approaches, give evidence based on human subjects to show there is room for improvement, and present some simple ideas that give better performance.

1 A Domain for Adaptive IR

As the heralded "information superhighway" continues to expand, the need for useful methods of navigating it becomes greater. There are already a number of applications that aid the user. These include the WebWatcher at CMU[4], the LIRA system at Stanford[1], and the Webhunter and Letizia projects at the MIT Media Lab[6]. The Softbot project at UW[2] and the FAQ Finder at the University of Chicago[3] both make use of Internet resources to answer natural-language user queries.

The large amounts of new data that appear daily on the Internet bring a number of challenges. This data cannot be easily processed by humans. Hence, relevance judgments, such as those provided in the Tipster collection[7], will not be readily available. More data means more documents, which means more words and terms to analyze. We will need some way to pick out the terms and phrases that carry meaning. Also there is a large amount of data that is essentially meaningless. Taking these documents into account will increase the system's search space. Since we cannot rely on human "experts" or oracles to help us, we will need to have a system that learns on its own.

We have focused on one part of the Internet, namely the USENET postings. This is a source of a large number of documents, and there is always new data available for training and testing. It is almost entirely text-based, so we don't have to worry about removing HTML commands or interpreting .gif files. There is a large variety of subjects covered, so we can study a particular area or more general topics quite easily. Most importantly, the postings are grouped together by category. This gives an automatic labelling of "relevance". An interesting point about these topics is that they are fairly broad. In studying these groups, we attempt to learn general concepts that can include many specific subjects. For example, the group comp.ai could include postings on expert systems, neural nets, and natural language. These may share none of the same terms, but they are all "about" AI.

Newsgroups have been studied in other projects. Mostly there has been work done in filtering. The SIFT project at Stamford[9] has users submit profiles for their interest. It then uses new documents as queries on the profiles. At CMU, the Newsweeder project[5] has users rate new documents in terms of relevance, and attempts to learn these ratings for classifying future documents. There are other interesting applications in this domain. The inverse of filtering is classification: given a document to be posted, determine the most appropriate group for it. We could also design an automated moderator that analyzes a message for content and relevance to a particular group.

2 Learning is Needed Here

In the domain of USENET postings, documents that come from the same group are presumably on the same topic. Hence, a user interested in one document from a group would want to retrieve others from that group. We wanted to see how well current systems (that do not try to learn) will perform on this kind of task. Our system of choice was SMART, developed by Salton[8].

We created collections of documents drawn from various subsets of the USENET hierarchy (minus their headers). SMART converts each of these documents into a vector whose features are the terms in the posting (removing stopwords like "the", and stemming). Given a particular document, we used it as a query to SMART. The system calculates the similarity between the vector for that document, and each of the other document vectors in the collection. The documents are then returned in decreasing order of similarity.

In a perfect world, the system will retrieve all of the documents from the target group. SMART, however, returns a fixed maximum number of documents to a query. The number of documents we want varies amongst groups. A fixed threshold will hurt the recall for large groups, and hurt the precision for small groups. So we set a different line for each group. A query's threshold is simply the size of its group.[1] Note that this is the value where recall is exactly precision; hence we have only one measure to maximize.

We chose our collections to cover different breadths and depths in the USENET hierarchy. Our first experiments drew 2000 documents chosen at random from the rec groups. Rec covers a broad variety of subjects, from antiques to comic books to recipes to woodworking. The subgroups go down to depth four in some cases (such as rec.arts.startrek.current, for example). We then examined the misc hierarchy which has fewer newsgroups, but still has a large number of postings. This collection gave us larger "clusters" of documents; newsgroups had more representatives here than did those in rec. We took all of the documents from the comp.ai hierarchy. While this was our smallest collection, it was selected due to expected overlap between document topics. We also chose all of the available documents from jhu.cs. This hierarchy is notable since there is no subgrouping. Our final and most studied collection is a potpourri of 100 messages each from six different newsgroups:

- misc.health.diabetes
- sci.math.num-analysis
- dc.politics
- rec.food.restaurants
- alt.tv.seinfeld
- comp.sys.ibm.pc.games.sports

These were chosen by hand to be semantically different. Also, being from different hierarchies, there would be minimal chance of crossposting. Each of the collections we used is detailed in Table 1.

In a run, each document in the collection was used as a query. We calculate the recall of the retrieval as described above. The second column in Table 1 gives the average recall over the entire run for each group of postings. SMART never does better than fifty percent recall, even on the group of semantically different documents. As the number of represented newsgroups goes up, accuracy diminishes.

3 How Humans Do

It is possible these poor results are pathological to our domain. Maybe newsgroup postings have too little semantic content. However, there is a machine that does manage to deal appropriately with postings – humans. They manage to send their postings to groups that are (in most cases) relevant. The performance of human beings would give us an approximate upper bound for our problem. So we devised an experiment to compare SMART to people.

We took our potpourri collection, and added 100 documents each from four other newsgroups to bring our total to 1000 messages in 10 newsgroups. We ensured that each of the postings had a non-empty body. The new newsgroups were again chosen to be semantically distinct from themselves and the other six.[2] In each trial, a subject was shown two of the thousand postings (minus headers). With probability 0.5, these would be from the same group. The subject's task was to determine if the postings were from the same group or not. The subjects were told that the ten groups were semantically different, but were not given the names of the newsgroups.

In Table 2, we see how seven subjects performed on this experiment. In total, 220 of the 258 presented pairs (85%) were correctly classified. Of the thirty-eight mistakes, thirty-six (95%) were on pairs from the same group. Ten of these were from talk.philosophy.misc, a group with a more eclectic range of covered topics. The abstract nature of the posts may have caused difficulty for the subjects.

The task we set out for humans is different from our original experiments with SMART. Hence we needed to run a similar experiment with the computer to make an

[1] We eliminate the problem of cross-posting, by ensuring that a document is posted to exactly one group.

[2] These new groups were:

- rec.arts.comics.dc.universe
- sci.military.naval
- talk.philosophy.misc
- humanities.lit.authors.shakespeare

Newsgroup	Accuracy	Coll Size	# of Groups
rec	0.148882	2000	405
misc	0.207153	2003	99
jhu.cs	0.327420	1289	14
ai	0.374909	530	13
potpourri	0.424781	600	6

Table 1: Collections we used. The second column gives SMART's average recall.

Subject	Correct	Attempts
A	43	51
B	6	7
C	36	40
D	31	40
E	33	40
F	40	40
G	31	40
Total	220	258

Table 2: Accuracy of human subjects on similarity task

accurate comparison. We want the system to decide if two postings belong in the same group. We also want to ensure that the computer sees pairs in about an equal ratio as our human subjects – 50% the same, 50% different. Of course, the collections used by humans and SMART should be the same.

We begin a trial in our human experiment by choosing a group at random, and a query document from it also at random. Next we build a collection by taking all of the other postings from the target group (i.e. 99 of them), and an equal number drawn at random from the other nine groups. We use our query document to rank the 198 other documents. Those in the top ninety-nine are claimed to be from the same group as the query, and those in the bottom half (or not retrieved at all) are claimed to be from different groups.

We ran 100 of these trials for a total of 19800 comparisons. Of these, 12684 were classified appropriately for an accuracy of 64%. This is worse than the humans did, so there is obviously much room for improvement on this task. While humans had almost all of their problems with same-group questions, the computer's errors were more evenly distributed. 3768 out of the 7116 incorrect classifications (53%) were on same-group questions.

It is interesting to compare how humans and SMART fare on the same-group pairings. Table 3 gives the human error rate and SMART's error rate on the ten categories. In two cases (the philosophy group and the politics group), SMART actually outperforms humans. We've already hypothesized why people do poorly on the philosophy postings. In the politics group, 55 of the 100 postings contain the word "whitewater". SMART can utilize this information, since it sees all of the politics posting. A person who sees one or two pairs in a run will not necessarily know this. However, in most of the groups, humans perform better than the computer. The difference is greatest in the groups devoted to comic books, Seinfeld, and Shakespeare. These postings contain a variety of proper names. A person can distinguish these as important and can cluster them by category. We know Hamlet and Othello are Shakespearean plays, but to SMART, they are two unrelated terms.

4 Cluster Analysis

This suggests that messages on the same general topic do not necessarily contain the same words. We may need to extract a set of words from our training data that appear to characterize the subject. One approach is to see what documents match well with others from the same group. These documents may contain the terms appropriate to the topic.

We clustered our original potpourri collection (with six newsgroups) to reflect how often each document was retrieved. For a particular posting, we counted how many times it was retrieved by query documents from its own group, and how many times it was retrieved by each other group. Remember that "retrieved" means that it appeared in the top n returned documents, where n is the group size. Here each group is of size 100. So, if SMART was performing perfectly, a query would have a score of 100 for its own group, and 0 for each other group.

We divided the queries into twelve different clusters. To do this, we first rounded each of the six numbers to 0 or 100, whichever was closer. The score for the document's actual group was moved to the first position. The scores for the other five groups were then sorted. The twelve clusters formed in this way measure whether (a) a document was recalled a majority of the time by its own group, and (b) how many other groups also retrieved it a majority of the time.

Three clusters have obvious important meaning. The

Pair Type	Errors	Chances	Perc	SMART Perc.
dccomics	3	13	23%	50%
diabetes	2	8	25%	28%
naval	2	11	18%	38%
num-analysis	2	15	13%	36%
philosophy	10	15	67%	41%
politics	5	14	36%	28%
restaurants	4	10	40%	44%
seinfeld	4	21	19%	47%
shakespeare	1	13	8%	34%
sports games	3	15	20%	37%

Table 3: Error rate comparison between people and SMART

cluster (100 0 0 0 0 0) is the optimal one. These documents are recovered often by their own group, and rarely by others. The cluster (0 0 0 0 0 0) represents postings that are rarely retrieved by anyone. (100 100 100 100 100 100) is a cluster of documents that match well with all of the other groups. These clusters are useful to study for error analysis.

Table 4 gives the distributions of the six hundred postings into the twelve clusters. The largest cluster is, fortunately, the optimal one. The next largest is the all-zero cluster. Notice that it is quite rare for a document to not match its own group, but to match well with other groups. Most of these 17 documents are from a thread in the restaurants group that discusses foreign relations! So SMART has done well in noting these postings are in an inappropriate place.

Unfortunately, we cannot say the same for those that are retrieved by none of the groups. There are some postings that are irrelevant to the topic, such as advertisements that were probably posted to a wide variety of groups, or "flames" that simply have name-calling. However, the majority of posts do have content. Most are very short, so there are a few words to match with other posts. Some are tangentially related, such as a discussion of hemophiliacs on the diabetics group, or arithmetic coding in the numerical analysis group. 45% of the restaurant group's postings are in the all-zero cluster more than any other. One possible explanation is a thread on potential restaurant names. These are phrases which a human could possibly identify as food-related, but a computer cannot. Throwing out all 188 of these problematic documents slightly raises the overall accuracy (in this case, from 42% to 56%). However, we cannot simply exclude them as irrelevant.

A possible bright spot is that we also have a collection of "good" documents. These are the postings in the optimal cluster. They represent the topics of the group. So we could try to use these to model the group. When we examine new postings, we could compare them to our previously discovered optimal set. A posting would retrieve optimal posts from its own group. The computer would decide that the posting belongs in the most represented group.

We ran an experiment to see how much performance improves. We made a new collection drawn from the same six groups. To maximize our test set size, we took all of the available documents in each group. This gave a total of 888 query documents. All of them were different from our original 600 queries. Also different groups had different numbers of representatives. Table 5 shows the result of using the 888 documents as queries on themselves (i.e. the same experiment as on the original potpourri collection). The average recall here is 44%[3]. It also gives the result of using these documents as queries on the 280 previously discovered optimal postings. The average recall here is slightly better, but it is still only 47%.

A problem here is that we are considering our optimal documents individually. Suppose there are two keywords A and B associated with a group. Some of our optimal documents will have A in them; let's assume 50%. The other half contain keyword B. A new document with just one of the keywords in it will extract only half of the optimal documents, i.e. the ones with its keyword. To rectify this, we will have to make sure that our representatives have both A and B in them.

A trivial way to do this is to combine the optimal postings into one macrodocument for the group. We refer to this technique as *pooling*. So now each group has exactly one representative in the collection. Given a new posting, we determine which macrodocument it is most similar to. We performed an experiment using the 888 new queries on the six macrodocuments formed from our optimal postings. The average accuracy in

[3]For this and all other experiments with the test set, all reported averages are weighted by the number of representatives per group.

Cluster						# of Postings
0	0	0	0	0	0	188
0	0	0	0	0	100	13
0	0	0	0	100	100	3
0	0	0	100	100	100	1
0	0	100	100	100	100	0
0	100	100	100	100	100	0
100	0	0	0	0	0	280
100	0	0	0	0	100	55
100	0	0	0	100	100	25
100	0	0	100	100	100	15
100	0	100	100	100	100	7
100	100	100	100	100	100	13

Table 4: Cluster distribution for the potpourri collection

Group	Recall on test set	Recall on good documents
Diabetes	0.53	0.52
Num. Analysis	0.33	0.44
Politics	0.39	0.34
Restaurants	0.32	0.49
Seinfeld	0.47	0.48
Sports Games	0.47	0.61
Average	0.4412	0.4743

Table 5: New documents as queries on themselves and on documents from the optimal cluster

this case is 79%.

We do need to ensure that pooling does help. If only one document in the macro is useful for a particular query, that document may be ranked highest when compared to the query alone. So we could simply classify via the group of the most similar document. To test this, we compared our new collection of 888 documents with our old collection of 600 documents. Now we just retrieve one document from the smaller group. The average accuracy in this experiment was 71%. Pooling does improve the results.

While pooling improves accuracy, we want to make sure we choose the correct documents to form each macrodocument. We described how we filter out documents that we hope are less useful. There are other methods of choosing the postings to pool. We tested three other ways. Our rounding method to find documents in the (100 0 0 0 0 0) cluster used a threshold level of 50 in each coordinate. Raising that level to 60 (i.e. the document was retrieved by 60% of the documents in its group) decreases the size of the optimal cluster. When pooling these together, average accuracy went up slightly to just above 80%. It is interesting to note here that almost three-quarters of the documents fall in the all-zero cluster. If we raise the threshold to 75, 98% of the documents are in the poor cluster.

Another potential method is to do random pooling. We chose ten random documents from each of the six groups, and formed macrodocuments from those. This collection had an average just under 80%, less than a percentage point increase relative to the trials with our original optimal postings. In both of the last two cases, the improvements resulted from a +10% increase in the score for the politics group. In the other cases, recall stayed the same or was lower than in our original method. However, the large size of the improved group drove the weighted average up.

The winner, though, was the third extra method we tested. We formed macrodocuments using all of the documents in the training set. In this case, the average accuracy on this classification task was 83%. There was improvement in four of the six groups, with only one or two percentage point decreases in the other two. Table 6 tabulates the results of all of our classification experiments.

We can also apply pooling to our experiments with filtering. We observed in Table 5 that using documents from our test set as queries onto the test set resulted in recall of 44%. As you might expect, using documents from the training set (the original 600 postings) gives a

Pooled Documents	Average Accuracy	Accuracy by Group					
		Dia	Num	Pol	Res	Sei	Spo
All training docs	0.8386	0.89	0.90	0.76	0.95	0.74	0.89
Optimal cluster docs Threshold: 60	0.8006	0.82	0.85	0.79	0.86	0.71	0.88
Random training docs	0.7962	0.85	0.90	0.71	0.88	0.72	0.83
Optimal cluster docs Threshold: 50	0.7898	0.85	0.89	0.60	0.96	0.72	0.91
No pooling; classify based on most similar doc	0.7052	0.78	0.78	0.49	0.74	0.71	0.80

Table 6: Classification accuracies with new test set

worse recall - 41%. However, if we use the pooled documents formed from the "optimal" cluster in training[4], recall rises to 71%. This is actually better than using the naive pooling that did so well on the classification task. If these macrodocuments are used as queries on the test set, we observe 69% recall. We summarize these results in Table 7.

5 Conclusions and Future Work

We now summarize the important points from our work.

- Newsgroups are an interesting real-world domain for IR research. They have a large amount of data which is already "labelled" by topic. These topics are broad enough to make simply keyword searching ineffectual.
- People seem to handle newsgroup postings well. Messages seem to be posted to relevant groups. On a similarity task, people do much better than SMART than identifying whether postings are from the same group. Hence we are optimistic that improvement in this domain is possible.
- Our clustering analysis shows that most postings do not match well with any group. Almost all of the documents are retrieved fewer than 75% of the time by their own group.
- We described a technique for creating macrodocuments to represent a newsgroup. Since posting labels come for free with the Internet, no human intervention is necessary to create the models or perform the classification. This method provides better classification accuracies on a test set than using single documents. Our results indicate, however, that attempting to intelligently select documents from the training set does not improve accuracy. Pooling documents also helps the filtering task, and we observe that choosing "optimal" postings does seem to improve recall.
- This suggests a method to retrieve documents similar to a query. First, we find the macrodocument that best represents the topic of the query. We then use the pooled document to perform the retrieval task.

The high classification percentages we report may be an upper bound for this problem. We plan to run another human experiment, this time giving subjects the names of the newsgroups, and asking them to classify postings. We suspect that humans would do very well here.

The success of the macrodocuments formed from the entire set may be attributed to the fact that such documents cover a large number of words. Removing the irrelevant words may improve accuracy. Our filtering techniques may have identified relevant postings, but these assuredly also contain irrelevant words. Also, the newsgroups in our pooling experiments were semantically distinct. Hence, it is unlikely that relevant words for one group would appear in other groups' macrodocuments. An experiment on a collection with more overlap may not be as successful. An algorithm that does filtering at the term level (not the message level) may be necessary.

Another possible weakness in pooling is that certain threads of discussion could be prevalent at training time. Hence, we cannot simply pool everything, since that would unduly weight these "fluke" subjects. So we will have to use filtering to weed these threads out. Also, this training will need to be continuous; otherwise we will not be able to tell which subjects are fads, and which are integral to the group. Hence, an adaptive system will be necessary.

If we are successful, we will be able to create a set of words that captures the topic of a newsgroup. Since newsgroups are sufficiently general, we will have keywords for a broad subject area. We can then apply

[4] with threshold 50

Pooled Documents	Average Recall	Recall by Group					
		Dia	Num	Pol	Res	Sei	Spo
Optimal cluster docs Threshold: 50	0.7083	0.79	0.70	0.65	0.59	0.73	0.79
All training docs	0.6867	0.75	0.67	0.66	0.53	0.72	0.79
No pooling; queries are docs in test set	0.4412	0.53	0.33	0.39	0.32	0.47	0.47
No pooling; queries are docs in training set	0.4117	0.52	0.34	0.41	0.28	0.46	0.46

Table 7: Recall from new test set using different macrodocuments as queries. Observe that the best result occurs by pooling documents from the optimal cluster.

these models to expand queries in searches in other domains.

References

[1] Shoham Y. Balabanović, M. and T. Yun. An adaptive agent for automated web browsing. *Journal of Image Representation and Visual Communication*, 6(4), December 1995.

[2] O. Etzioni and D. Weld. A softbot-based interface to the internet. *Communications of the ACM*, July 1994.

[3] Burke R. Martin C. Hammond, K. and S. Lytinen. Faq finder: A case-based approach to knowledge navigation. In *Proc. 1995 AAAI Spring Symp. on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995.

[4] Freitag Joachims, Mitchell and Armstrong. Webwatcher: A learning apprentice for the world wide web. *Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995.

[5] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, Tahoe City, CA, July 1995.

[6] Metral M. Lashkari, Y. and P. Maes. Collaborative interface agents. In *Conference of the American Association for Artificial Intelligence*, Seattle, WA, August 1994.

[7] J. Prange, editor. *TIPSTER Text Phase II*, 1995.

[8] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[9] T. Yan and H. Garcia-Molina. Sift - a tool for wide-area information dissemination. *Proceedings of the 1995 USENIX Technical Conference*, pages 177–186, 1995.