Combining Lexicon Expansion, Information Retrieval, and Cluster-based Ranking for Biomedical Question Answering

A. M. Cohen[1], J. Yang[1], S. Fisher[2], B. Roark[2], and W.R. Hersh[1]

Department of Medical Informatics and Clinical Epidemiology[1],
Department of Computer Science and Electrical Engineering, OGI School of Science and Engineering[2]
Oregon Health & Science University, Portland, OR, USA

ABSTRACT

The Oregon Health & Science University submission to the TREC 2006 Genomics Track approached the question answer extraction task in three phases. In the first phase the biological questions were parsed into relevant entities and query expressions were generated. The second phase retrieved relevant passages from the corpus using Lucene as an information retrieval engine. The third phase performed ranking of the retrieved passages and generated the final submitted output. Through these experiments and comparison with the approaches of others we hope to learn the contribution and value of several techniques applicable to question answer extraction including: lexicon-based query term expansion, query back-off techniques for questions with few applicable passages, and passage clustering for identifying distinct aspects of question answers. Our experiments showed no improvement after cluster-based ranking. Maximal span based passage indexing proved to be too coarse, resulting in an overall average performing passage MAP of 4%.

1 INTRODUCTION

The 2006 Text Retrieval Conference (TREC) Genomics Track consisted of one main question answering task, which was scored by three measures. The question answering task used 28 biomedical topics from the ad hoc task of the previous year. These topics were rephrased into questions, and the challenge was to extract passages answering each question from a large corpus of over 160,000 biomedical full text articles selected from journals known to publish papers on genomics research. The answers for each topic were pooled and then evaluated by domain experts to create a gold standard. Each submission was scored on all the topics for document mean average precision (MAP), character-based mean passage precision, and aspect precision.

2 BACKGROUND

Effective text processing of the biomedical literature can be a useful aid to biomedical researchers (Cohen and Hersh, 2005, Kerns *et al.*, 2005). However, to provide benefit, tasks addressable by text processing must be identified, approaches that give good results must be found, and end user systems that use these approaches must be made available. One of the most common uses of the biomedical literature by scientists is to determine what is currently known about a subject that they are interested in studying. This kind of information need can be framed as a question answering task, where the question is in a form like, "What effect does gene X have on the etiology of disease Y". Good answers to these questions not only address the subject of the question, but cover the range of information known about the question, that is, the *aspects* of the question, which is the termed used in the track task.

The TREC 2006 Genomics track task attempted to emulate this question-based type of information need in a controlled, comparable form, using a specific full text literature corpus. The literature corpus consisted of 162,259 full text HTML articles, published between the years 1995 and 2006, and downloaded with the permission of the publishers from the Highwire Press web site (http://highwire.stanford.edu/). This literature corpus includes 49 journals that were known to publish articles on genomics subjects. However, the articles in the corpus included everything available from the web site, not just genomics articles.

Using the given literature corpus, the task was set up to be a extraction-based question answering task, Answers needed to come from contiguous passages of the literature corpus. Systems were to submit a ranked list of up to 1000 character-based passages for each topic. The character based passage specified the PubMed ID (PMID), starting offset in characters, and length of passage corresponding to the passage within the HTML file being nominated as relevant to answering the question.

Submissions were free to use any size submitted passage they desired, whether that be sentences, sentences fragments, paragraphs, etc. However, the maximum allowable passage was restricted by enforcing the rule that a passage could not include any HTML <P> or </P> tags. This effectively limited submitted passages to around one paragraph of text, although entire reference sections from the end of papers sometimes were included by this rule. A file of the maximum legal spans was provided by the track administrators.

A character-based gold standard set of answers was created from the submissions by using pooling, combined with expert judging. Submitted answers were mapped to their containing maximum legal span, and then were

pooled for judging by taking the top ranked spans from each entry until 1000 spans were collected for each topic.

Human judges with expertise in biology were then asked to rate each pooled span for relevance, and select the relevant answer text from a plain text version from the pooled HTML span. These plain text selections were then mapped back to the original HTML file using a string alignment algorithm to create the gold standard set of passages. Each gold standard passage also had assigned to it by the judges one or more MeSH terms, which were used to designate the various aspects of the question's answer. This was intended to separate answers into groups which reflected the kind of information that was available for answering the question.

The submission for a system consisted of a ranked list of up to 1000 passages for each topic. This ranked list was scored three ways, each measure being a variation of mean average precision (MAP).

The first measure was document MAP. This took the highest ranked passage for a document as the documents rank. The second measure used was character-based passage MAP, which measured the cumulative overlap between characters in relevant and nominated passages at each point of correct passage recall. The third measure was aspect-based MAP. This took the highest rank of a passage with a given assigned MeSH term as the recall rank for that MeSH term.

Having three separate measures allows a more detailed study of what algorithms and approaches are more suitable for the various parts of a question answering system. While good document-based MAP is useful, full text papers can be long and time consuming to read. Character-based MAP allows users to focus their reading and time on only the sections most likely to be helpful. Finally, aspect-based MAP measures the ability of a system to provide broad coverage of a topic. This also affects the user experience, since all else being equal, it would be preferable to have a shorter list of passages to that covered the material rather than a longer list. These separate measures will enable us to draw some general conclusions about the effectiveness of the various techniques that can be used for biomedical question answering.

3 SYSTEM AND METHODS

The Oregon Health & Science University submission to the TREC 2006 Genomics Track approached the question answer extraction task in three phases. In the first phase the biological questions were parsed into relevant entities and query expressions were generated. The second phase retrieved relevant passages from the corpus using Lucene (http://lucene.apache.org/) as an information retrieval engine. The third phase performed ranking of the retrieved passages and generated the final submitted output. The overall architecture of our system can be seen in Figure 1.

3.1 Topic Parsing and Query Generation

The goal of the first phase of the processing was to transform the topic question into a Boolean expression for processing by an information retrieval engine in the next stage. This was done in three steps. The first step parsed the topic text into a set of relevant string entities and entity types, the second step expanded entities with synonymous terms, and the third step created a Boolean query expression from the resulting lists of terms.

Parsing the topic question into relevant entities was done using a set of hand crafted regular expressions. Essentially these expressions had "slots" for relevant entity types such as genes and proteins, diseases, biological processes, tissues, and organs, and included surrounding text that set the context for these entity types. Since there was no substantial set of question training data, these regular expressions were created based on looking at the topic questions themselves, and therefore they have not been studied as to their accuracy and comprehensiveness as far as extracting the relevant entity types. Within each topic the question phrases followed a formulaic structure, and we expect that similarly structured questions would be adequately parsed as well. However, this is not a general approach to biomedical question parsing, and certainly much more work is needed in this area.

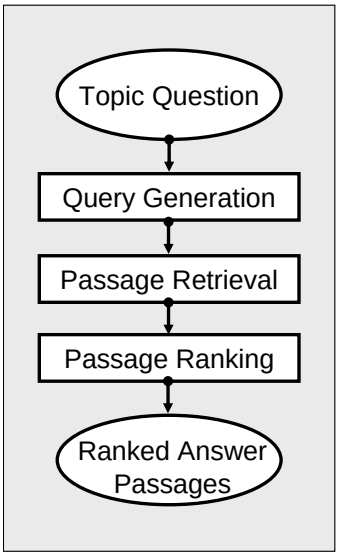

Figure 1. Question Answering System Architecture

The entities extracted from each query topic were then expanded in an entity type-specific manner. Genes and proteins were expanded with a string matching based method using a thesaurus constructed from the Entrez gene database (http://www.ncbi.nlm.nih.gov/entrez/). First a search for an exact match was attempted, if that failed, gene/protein names were split into components based on parenthesis, and the components searched for in the thesaurus. Finally, orthographic variants of synonymous gene names were generated by replacing hyphens with spaces (and vice-versa) and separating runs of alphabetic characters from runs of numeric characters by hyphens and spaces.

Entities of the types diseases, biological processes, tissues, and organs were similarly expanded using a thesaurus created from the MeSH controlled vocabulary (http://www.nlm.nih.gov/mesh/meshhome.html). Synonyms that included a parenthetical expression were separated as separate terms. Two part phrases separated by commas were split and put back together inverted as additional synonyms.

The query topics each had a designated generic topic type (GTT) which gave some indication of the domain of the question. The information for GTT 5 was used when building these queries by including special GTT-specific terms for mutations and alterations.

Lastly, some regular expressions included slots for topic-specific words. These were also included in the query, and were expanded in form by reducing the word to its root and applying several common suffixes such as "-ing", "-es", "-ed" and "-ation".

Each of these lists of synonymous terms was combined into a single Boolean query expression in the following manner. Each individual list of synonymous terms was combined into a single expression by concatenating the terms with "OR" in between each pair of terms. Then the OR expressions for all of the different topic entities were combined by placing parenthesis around each OR expression and concatenating these expressions with "AND" in between each pair of terms. This final query expression was then passed to the passage retrieval engine.

The construction of the query intentionally includes synonyms and automatically generated variants that may or may not be relevant to the question. Many of the topics were found to have very few hits when a Pubmed search was performed using just the extracted the entity terms from topic question. Therefore high recall was favored over high precision in query creation.

3.2 Passage Retrieval

The second phase consisted of several steps. The first step was to parse the given corpus of medical documents into legal spans, and then index the resulting set of legal spans using the open-source text retrieval engine, Lucene (http://lucene.apache.org). This indexing process only has to be done once, and is not query specific. Once this initial indexing was finished, we used queries built in the first phase (see 3.1) to retrieve candidate answer passages from the indexed set of passages. If the fully expanded query did not return the maximum allowed 1000 passages per question, a two-step back-off strategy was used to find more documents related to the question. The result of the initial query and two back-off steps was up to three sets of passages for each question. All three sets were passed to the third phase (see 3.3), where they were clustered independently of each other. In our submission, all members of the first set were ranked higher than any of the second set, and all members of the second set higher than those in the third.

The initial parsing of medical texts into legal passages was achieved using the legal spans character offsets provided at the official genomics track website. We created a new file for each legal span, noting which medical document the span originally came from. We thus had a very large set of mostly rather small files. This large set of passages was indexed using the basic facility provided by Lucene, which efficiently stores the term frequencies of each file being indexed. In this case, a "term" is simply a white-space delimited string.

Once we had the Lucene index, we could perform any number of queries very quickly. The expanded queries from the initial phase were used directly to retrieve the most relevant passages for each question. With each passage returned, a TF-IDF score is also provided by Lucene. We did not filter out identical passages from different sources. Unfortunately, even with the query expansion, there were often fewer than 1,000 passages retrieved (in some cases, zero passages). One reason being that the queries impose hard constraints, and most of the passages were small and considered in isolation. Thus, even if two consecutive passages considered together contained all the query words, neither would be retrieved.

We implemented a back-off strategy to address the problem just described of not finding enough passages. Basically, we used the same expanded queries, but relaxed the AND constraints. There were two back-off implementations. The first was to find all of the original documents that satisfied the expanded query, and then from that resulting set of legal passages, find passages that had the maximum number of the different query word sets represented. That is, if there were three sets of expanded query words in an expanded query, meaning there were two AND statements, then find passages with two of the sets represented (as opposed to all three sets represented, which is what the original query did). If the

addition of this new set of results still did not bring the total number of legal passages to 1,000, then we resorted to the second back-off implementation. This consisted of running the query with all AND statements changed to OR statements against the entire set of passages. For each of the back-off implementations, all passages that had been retrieved in an earlier stage were filtered out.

System speed using the retrieval approach as described is acceptable. Indexing all of the legal passages took less than a day on a 2.7 GHz x86 processor system running Linux. Running a single query took about one second.

3.3 Cluster-based Ranking

The final phase performed ranking of the retrieved passages. Three strategies were applied that were intended to raise the ranking of relevant documents with differing aspects. Cluster-based ranking, at two settings of cluster quality, first clustered similar passages by word vectors and then ordered passages round-robin between clusters and high to low within a cluster based on retrieval TFIDF score. We also submitted a run simply ordering passages based on retrieval TFIDF score.

The passages from each retrieval step were modeled as word vectors after removal of stop list words (from ftp://ftp.cs.cornell.edu/pub/smart/english.stop) and Porter-stemming (Porter, 1980). Let M be the number of distinct words in our collection of N passages, passage i is represented as vector $P_i = \langle w_{i1}, w_{i2},\ldots,w_{iM}\rangle$, $w_{i,j}$ is the frequency of jth word in the ith passage. Hence, the retrieved passages were represented as a NxM matrix to the clustering algorithm, with each passage being a row in the matrix.

We used a discrete k-means algorithm implemented in CLUTO (available for download at http://www-users.cs.umn.edu/~karypis/cluto/index.html) in our clustering process. The implementation allows some manipulation of parameters. For example, we chose the popular cosine measure: Cos(Pi, Pj)=Pi ● Pj/|Pi| |Pj| to calculate the similarity between passages. We also normalized each passage (row) to the highest word frequency of that passage (maxtf) to adjust the weight of individual words to account for the length of the passage.

The number of clusters is the important most parameter that CLUTO takes in as user input. We decided this number by searching for a point where the improvement too cluster quality gained by increasing the cluster number by one tapered off. Let ε be the ratio of improvement of internal similarity of all cluster by increasing the number of cluster by one:

$\varepsilon = [I2_{(n+1)} - I2_{(n)}]/I2_{(n)}$ where I2 is the measure of internal similarity for all clusters obtained from CLUTO output, and n is the number of clusters.

We first ran the clustering algorithm on our retrieved passage matrix, calculating ε for each increment of n. When the condition $\varepsilon < \varepsilon_0$ was reached, we took n as our input for number of clusters for our true run. ε_0 was a constant set at 0.01. We decided ε_0 by running our sample data and choosing a value that resulted in about 20-50 passages in each cluster.

The clustering algorithm arranged the collection of passages in n clusters according to their similarity. The passages were then reordered round-robin among clusters and from high to low within a cluster based on retrieval TFIDF score.

Since we had a two-step backoff strategy, the results from initial run, first backoff and second backoff were clustered and reranked separately to preserve the original priority. This was to ensure that passages from initial run were always ranked higher than the two backoff runs, and first backoff higher than second backoff. Furthermore, if less than ten passages were return in the retrieval step; we presented these in the original order, without reranking.

Input cluster size was an important parameter to the reranking algorithm. To study how cluster size would influence performance, we also submitted a run with ε_0 set at 0.05, which resulted in bigger or looser clusters. In our third run, we simply ordered passage according to their retrieval TFIDF score, without reranking.

4 RESULTS

Performance was determined by the official scoring program for the track, trecgen2006_eval. The results of our three runs are presented in Table 1. For comparison, the high, low, and mean scores for all runs submitted to the track are shown for each of the three measures. Table 2 shows a comparison of our passage MAP against the track median score by the number of hits our OHSUCluster system had for that query topic.

RUN	DOCUMENT MAP	PASSAGE MAP	ASPECT MAP
OHSUNoClu	0.3274	0.0419	0.1946
OHSUCluster	0.3042	0.0344	0.1880
OHSUBigClu	0.3051	0.0379	0.1892
TREC Mean	0.2887	0.0392	0.1643
TREC Min	0.0198	0.0007	0.0110
TREC Max	0.5439	0.1486	0.4411

Table 1. System performance across all topics.

# QUERY HITS	TOPICS WITH SCORE >= MEDIAN	
	MANUAL+INTERACTIVE	AUTOMATIC
≥ 10	7/11	9/11
< 10	5/15	6/15

Table 2. Number of original query hits versus passage MAP score for OHSUCluster compared to track median score.

Performance for all three variations of our system on all three measures was just above average. The system without clustering performed moderately better than than either of the two clustering variations. Performance was far below the best systems for all three measures.

Figures 2, 3, and 4 present the performance of our best performing system, OHSUNoClu, per query topic, and compared to the best scoring run submitted to TREC for that topic. The dark bottom portion of the bar is the OHSUNoClu score, and the total height of the bar is the TREC best score. The light portion of the bar represents the difference between our OHSUNoClu system and the best submitted system for that measure and that topic.

5 DISCUSSION

Several things are made clear from Table 1. First, the clustering approach did not improve performance for any of the measures. Both large and small clusters reduced performance somewhat. Most surprising is that the clustering did not improve aspect level performance. Increasing the variety of top ranked passages in order to improve this measure was the primary intent of the clustering step. More investigation is needed to determine why this did not have the desired result. The method of aspect performance evaluation removed duplicated aspects from the ranked list before scoring. This may have decreased the influence of novelty on the aspect scores. Another possible explanation for aspect performance could be that the clustering was dominated by terms not related to the expert judge assigned aspects. One possible improvement would be to cluster passage that include query terms based on the MeSH terms assigned to the journal article, instead of the individual words in the passage.

Second, both passage and aspect performance tends to be highly influenced by document performance. Lastly, even in the best scoring systems, isolating out relevant text from non-relevant surrounding text was a difficult, task. Of the characters returned by our system, only about 4% contributed to the relevant answer. Even for the best system, only about 15% of characters contributed to the relevant answer text.

As can be see in Figure 2, document retrieval performance was generally quite good. However, for several of the topics (164, 166, 170, 171, 175, 177, and 178) essentially no relevant documents were retrieved. This could be a result of a failure of the primary query, combined with a lack of precision of the back off queries. Whether the primary query failed because of a lack of sufficient synonyms requires more investigation. Certainly there were topics that had few relevant passages in the corpus, however, other systems were able to find these, and therefore our word based IR approach needs to be augmented in some manner.

As can be seen in Table 2, our system performance on the passage metric largely depended on whether or not the original query returned sufficient results. When the original expanded query returned 10 or more passages, the system scored as well or better than the median in the majority of cases, versus scoring above the median for under half the topics when the original query returned fewer than 10 passages. This indicates that our retrieval and back-off strategy could be improved.

One possible explanation for the weak results of the document search may be that our system indexed only the words in the maximal legal span and did not use any of the MeSH terms associated with the article. In fact, since MeSH terms were used to determine the relevant passage aspects, there is certainly a good possibility that they would be useful for aiding the document search as well. It is also possible that incorporating the distance between query terms would have improved our document ranking.

Figure 3 shows the character-based passage average precision compared to the best TREC score per topic. As expected, passage scores were low when document scores were low. However, scores are about half the maximum seen (among all track submissions) when document scores are high. This can be attributed to our rough granularity of using each maximum span as a passage. It appears that using a finer approach, for example, indexing individual sentences, may have resulted in better passage precision.

Figure 4 shows the results of aspect average precision. Again, scores are low when document MAP is low. However, when document MAP is high, aspect average precision performance is frequently very good, half or better of the maximum submitted system performance. This also could probably have been improved somewhat by finer granularity passages, for example using sentences or sentence pairs.

6 CONCLUSIONS

The TREC Genomics track question answer extraction task was a new one for this year, and pushed the envelope of genomics biomedical information retrieval. Feedback from the results of this year will be very useful in improving future work in this area. Modifications to our approach that could be advantageous include the use of MeSH terms when indexing the document corpus, finer level passage granularity using sentences or sentence pairs, including inter-term distance in query ranking, and clustering based on MeSH term mappings instead of individual words.

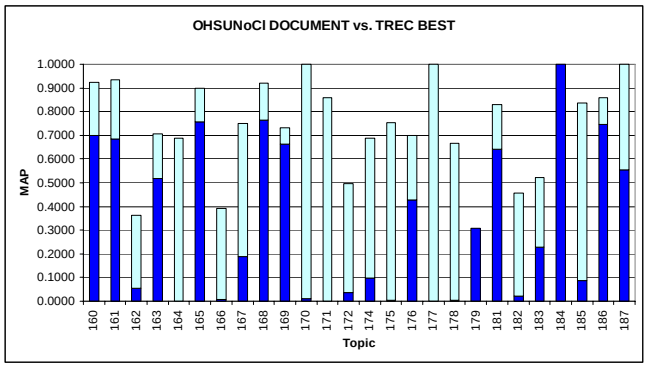

Figure 2. Document MAP performance per topic (dark), compared to TREC submitted best (light).

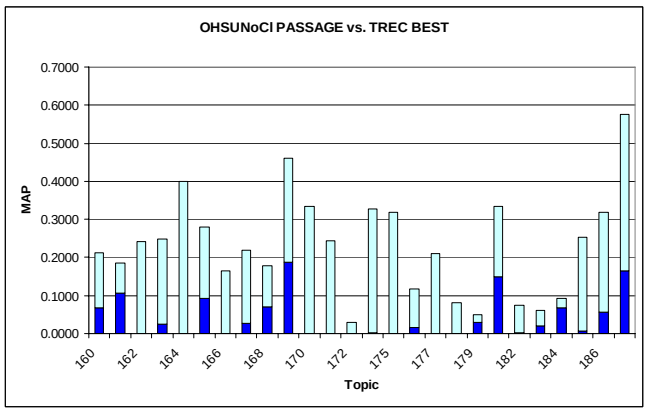

Figure 3. Passage MAP performance per topic (dark), compared to TREC submitted best (light).

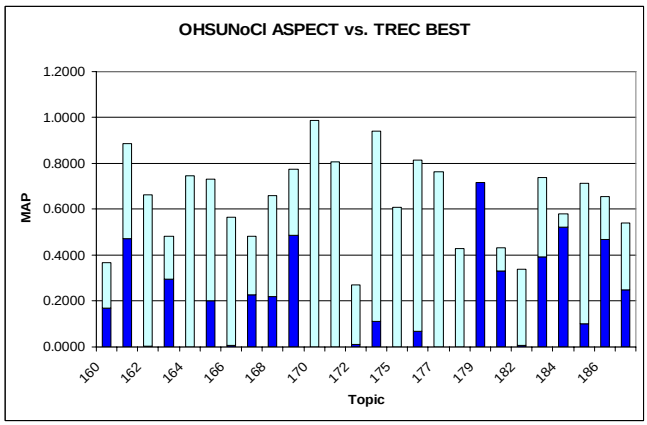

Figure 4. Aspect MAP performance per topic (dark), compared to TREC submitted best (light).

ACKNOWLEDGEMENTS

This work was supported in part by Grant ITR-0325160 from the National Science Foundation.

REFERENCES

Cohen, A. M. and Hersh, W. (2005) A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, **6**, 57-71.

Kerns, R. T., Ravindranathan, A., Hassan, S., Cage, M. P., York, T., Sikela, J. M., Williams, R. W. and Miles, M. F. (2005) Ethanol-responsive brain region expression networks: implications for behavioral responses to acute ethanol in DBA/2J versus C57BL/6J mice. *J Neurosci*, **25**, 2255-66.

Porter, M. F. (1980) An algorithm for suffix stripping. *Program*, **14**, 127-130.