UMass at TREC ciQA 2006

Giridhar Kumaran and James Allan
Center for Intelligent Information Retrieval
University of Massachusetts, Amherst

1 Introduction

The characteristics of the ciQA Track namely the short templated queries and the scope for user interaction were the motivating factors for our interest in participating in the track. Templated queries represent a new paradigm of information-seeking more suited for specialized tasks. While work has been done in document retrieval for templated queries as part of the Global Autonomous Language Exploitation[1] (GALE) program, the retrieval of snippets of information in lieu of documents was an interesting challenge. We also utilized the opportunity to try a suite of *minimally* interactive techniques, some of which helped and some did not. We believe we have a reasonable understanding of why some approaches worked while other failed, and contend that more experimentation and analysis is necessary to tease out various interaction effects between the suite of approaches we tried.

2 Baseline Runs

We used version 2.3.2 of the Indri search engine, developed as part of the Lemur[2] project as the foundation for creating our baseline QA system. While the inference network-based retrieval framework of Indri permits the use of structured queries, the use of language modeling techniques provides better estimates of probabilities for query evaluation. We used the query-likelihood variant of statistical language modeling. Given a query

$$Q = q_1 q_2 q_3 \ldots q_n,$$

and a document

$$D = d_1 d_2 d_3 \ldots d_n,$$

the probability $P(Q|D)$ that the query would be generated by the document is

$$P(Q|D) = \prod_{j=1}^{n} P(q_j|D)$$

with

$$P_{ML}(q_j|D) = \frac{c(q_j;D)}{\sum_{i=1}^{n} c(w_i;D)}$$

where $c(q_i;D)$ represents the number of times that term q_i occurs in document D and ML refers to maximum likelihood.

The pseudo-relevance feedback mechanism we used is based on relevance models [3].

Our QA system derived from Indri was composed of two stages. While the first stage converted the templated queries into Indri language queries and extracted relevant passages, the second stage worked at a lower level to identify relevant snippets. The snippets we found were primarily relevant sentences.

We now present the series of steps we followed to create our baseline QA system.

2.1 Stage I

1. *Phrase identification*: The first step was to identify phrases of length two in the query, both in

[1] http://www.darpa.mil/ipto/programs/gale/
[2] http://www.lemurproject.org

the fields of the template as well as in the narrative. To do this, we used the `expressionCnt` operator available in the Indri query language. By checking if every adjacent pair of terms in the query occurred in the corpus at least once within a term window of size three, we identified potentially useful bigrams. Phrases are known to be precision enhancing, and our intention was to use the identified phrases to retrieve vital nuggets. The identified bigrams were included in the query using operators that specified that the constituent terms must occur adjacent to each other.

2. *Narrative processing*: The narrative accompanying each templated query was an elaboration of the information need. However, the narrative often contained a number of terms that hampered retrieval instead of helping with it. Our simple attempted solution to this was to consider only terms in the narrative that occurred within a window of five terms, of terms that appeared in the template fields.

3. *Passage Retrieval and Processing*: With the exception of Template 2, for which the *goods* field was weighted twice as much as the terms in other fields, we weighted all the terms from the template and narrative equally to create a query. The query was further modified to retrieve passages of length 200 from the collection. Using the passage-specific query, we retrieved the top 10 passages. We observed from training data that often the passages retrieved were lead lines of documents, and contained a lot of extraneous information that contributed noise to the final output. To oversome this problem, we utilized the fact that lead lines are mostly capitalized, and created a simple filter that rejected a passage from consideration for further processing if more than 50% of the terms in it were capitalized.

2.2 Stage II

1. *Sentence segmentation*: Once we obtained the top passages from the previous stage, we split them into sentences using a sentence segmenter [4]. We discarded sentences of length less than 5 terms, and more than 50 terms. In the former case, the sentences conveyed too little information to be useful, while in the latter they contained too much extraneous information.

2. *Sentence selection and Processing*: We re-ran the original query created in Stage I, bereft of the passage modification, against our set of candidate sentences. Once we obtained the ranked list, we used each sentence in the ranked list as a query against all sentences lower in the list. By eliminating all the sentences that have a similarity more than a particular threshold, we removed potential duplicates and redundant information. Finally, we output the remaining sentences in the order they appeared in the ranked list until the character limit was reached.

UMass fielded two baseline systems, *UMASSauto1* and *UMASSauto2*. While *UMASSauto1* was the system described above, *UMASSauto2* differed in the way the query was created in Step 3 of Stage I. Using the original query, a USENET archive[3] was queried to obtain the top matching USENET newsgroups. Once this was done, the query was re-issued targeting the most frequent newsgroup that appeared in response to the previous query. The 100 documents that were retrieved were used as an external relevance model to expand the original query [2]. This expanded query was used further along the baseline system.

3 Interaction Runs

There has recently been great interest in utilizing annotations in data provided by the Information Extraction community for Information Retrieval. While we were unable to utilize in time annotations from the Automatic Content Extraction [4] program, we attempted a step towards that direction by making use of named entities identified using BBN Identifinder

[3] `http://groups.google.com`

[4] `http://www.nist.gov/speech/tests/ace`

[1]. To also move away from the usual attempt to identify named entities and fold them into the query in a different way, we decided to instead use them for post-retrieval snippet processing. This was motivated by the fact that we noticed that in the training data often unrelated people, organizations and locations were returned in the results. Since it is not possible to predict which named entities were unrelated, we decided to involve the user in making that decision, and use the information to *clean-up* the final results. To this end, as one of our interaction mechanisms, we provided the user with separate lists of people, locations, and organizations identified in the final output from our baseline run. Along with each named entity, we provided a sentence in which it occurred to provide some context to help the user make a decision whether the named entity was **not** relevant to the information need. We hoped that this kind of *negative* feedback will help improve the precision of our results.

The second interface mechanism we deployed was a simple spelling-correction system that was intended to take care of spelling variations in the corpus due to typos and cultural differences. For example, *estrogen* and *oestrogen* are accepted spellings in different countries, but no stemming algorithm places them to the same equivalence class. For each term in the query that had at most two terms in the corpus with a edit distance of one, we displayed the term along with the identified terms and asked the user to select what they considered true typos and alternate spellings. We believed that this information from the user could help improve recall in a few queries.

In our second baseline system we used USENET newsgroups as a model of external relevance information. However, the model was selected automatically. This provided another opportunity for interaction in which the user was provided the titles of USENET newsgroups from an initial retrieval along with a brief description. Given the description, the user was asked to select the newsgroup(s) they thought were most likely to discuss the query's topic. The information gleaned from this interaction had potential to improve both precision as well as recall.

While the first interactive run we submitted *UMASSi1* had all the above features, the second interactive run *UMASSi2* was different in that it used a combination of the user-selected newsgroups and the automatically selected ones for pseudo-relevance feedback.

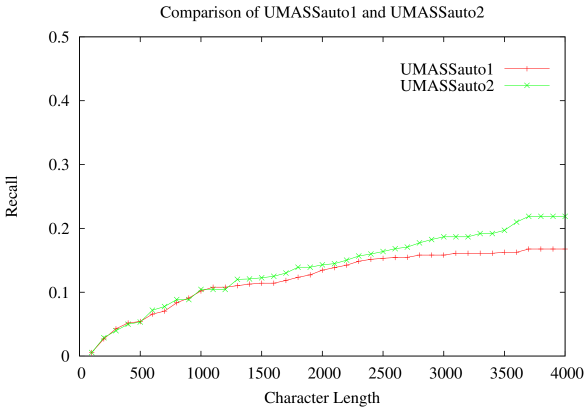

Figure 1: Comparison of the two automatic runs

4 Results

We now present the results of all our runs. Figure 1 compares the two automatic runs we submitted, *UMASSauto1* and *UMASSauto2*. We can observe that the second automatic run that used an external corpus for pseudo-relevance feedback outperformed the one that did not do so. The improvement in recall is more pronounced lower down the ranked list of snippets returned by the system.

Figure 2 provides a comparison between the better automatic run and the two interactive runs. The interesting thing to note is that both interactive runs performed worse than the automatic runs. This shows that users were unable to select the best news group for pseudo-relevance feedback. Even the interactive run that used a mixture of manual and automatic selection of news groups was not able to match the performance of the automatic run.

Table 1 contains the F-scores that each system achieved. As expected, *UMASSauto2* achieved a higher F-score. The fact that recall is weighted thrice

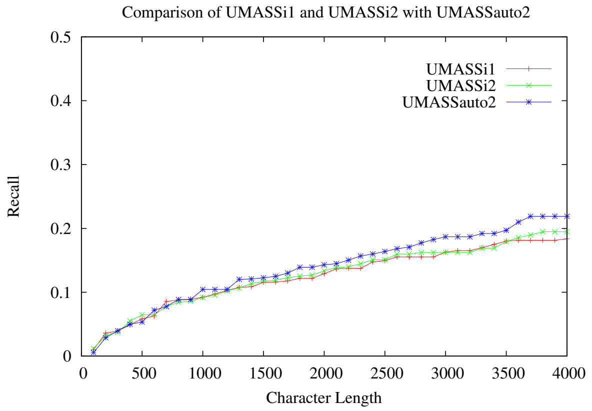

Figure 2: Comparison of the better automatic run with the two interactive runs.

	F-score
UMASSauto1	0.132
UMASSauto2	0.170
UMASSi1	0.150
UMASSi2	0.160

Table 1: Comparison of the F-scores for each run.

as more as precision in the formula to calculate the F-score further amplified the difference between the various systems.

Table 2 compares the Mean Average Precision (MAP) scores of the four runs. In contrast to the drop in recall, the precision of the interactive runs is higher. We hypothesize that this is the effect of using the negative feedback provided by the users in the form of non-relevant named-entity identification. However, the problems caused by the apparent failure of newsgroup selection seemed to have dampened the effect of named-entity processing.

5 Conclusions

The failure of one of the interaction mechanisms and the success of another led to mixed results for the interactive runs. We designed our interaction mechanisms keeping in mind that the user should be spared from expending too much effort in understanding and answering the questions. Analysis of the time taken by the users to complete the interaction forms showed that for almost 80% of the time the users took three minutes or more to complete their responses. This shows that we still have a long way to go before we achieve our goal of minimal interaction, and believe that the actual design of the interface too plays an important role.

	MAP
UMASSauto1	0.061
UMASSauto2	0.074
UMASSi1	0.074
UMASSi2	0.081 (0.067)

Table 2: Comparison of the MAP scores for each run. The values in brackets are p-values from a two-tailed paired t-test comparing the run with *UMASSauto1*

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

References

[1] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[2] Giridhar Kumaran and James Allan. Simple questions to improve pseudo-relevance feedback results. In *ACM SIGIR Conference*, pages 661–662, 2006.

[3] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *ACM SIGIR Conference*, pages 120–127, 2001.

[4] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Fifth conference on applied natural language processing*, pages 16–19, 1997.