Answer Models for Question Answering Passage Retrieval

Andrés Corrada-Emmanuel
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts at Amherst
Amherst, MA 01003-9264
corrada@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts at Amherst
Amherst, MA 01003-9264
croft@cs.umass.edu

ABSTRACT

Answer patterns have been shown to improve the performance of open-domain factoid QA systems. Their use, however, requires either constructing the patterns manually or developing algorithms for learning them automatically. We present here a simpler approach that extends the techniques of language modeling to create *answer models.* These are language models trained on the correct answers to training questions. We show how they fit naturally into a probabilistic model for answer passage retrieval and demonstrate their effectiveness on the TREC 2002 QA Corpus.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval Models*

General Terms

Algorithms, Experimentation

Keywords

Question Answering Systems, Language Models, Answer Passage Retrieval, Open Domain Factoid Questions

1. INTRODUCTION

Open-domain factoid questions, such as those used in the TREC QA evaluations, tend to have small passages of text as answers. To the casual observer, the text in these passages frequently seem formulaic. For example, "When was PERSON born?" can be answered by pieces of text like: "PERSON was born in DATE", "Born in DATE, PERSON ...", etc. Thus, we can use these patterns to eliminate or rerank candidate answer passages to a question. Coupled with a question classifier, answer patterns should improve the performance of QA systems.

This insight has been implemented successfully in QA systems. Sabboutin and Sabboutin used manually constructed *answer patterns* to construct a QA system that was succesful in the TREC 2001 QA track [4]. The drawback of manual construction of the answer patterns was removed by the work of Ravichandran and Hovy [3], who introduced an algorithm that used the Web to learn answer patterns.

We present here a third alternative for exploiting the appearance of answer patterns in factoid questions - *answer models.* These are language models that have been trained on the correct answers to training questions. As such, their training is fully automatic and does not rely on external resources to improve the performance of a QA system.

2. LANGUAGE MODEL FRAMEWORK

The basic approach to using language models for QA is to reinterpret the query likelihood algorithm [2] in terms of questions and passages instead of queries and documents. A quick derivation of this approach in a QA context is helpful since it will also show how answer models can be incorporated into a statistical QA system.

We want to rank answer passages by our estimate of the quantity, $P(a|q)$ where a is a candidate answer passage and q, a question. In practice, it is better to calculate the *query likelihood* - $P(q|a)$. Using Bayes' theorem:

$$P(a \mid q) = P(a)P(q \mid a)/P(q) \tag{1}$$

Since we are only interested in ranking passages for a given question, we neglect $P(q)$ and rank by the quantity,

$$P(a)P(q \mid a). \tag{2}$$

Query likelihood ranking assumes $P(a)$ is constant. The purpose of our experiments is to relax this assumption. We use *answer models* to estimate $P(a)$. Answer models are n-gram language models that have been trained on correct answers to questions. The training set of answers have been transformed to abstract away the particulars of specific words in the text. We do this by using BBN's IdentiFinder to tag the following named entities: PERSON, LOCATION, ORGANIZATION, NUMBER, DATE, TIME, MONEY and PERCENT. Any text tagged with these classes is replaced by a single token denoting its class. It is this abstracted text consisting of regular words and class tokens that is used, in the experiments quoted here, to build bigram models smoothed with absolute discounting.

Since we are using n-gram approximations for $P(q \mid a)$ and $P(a)$, we introduced two tuning parameters. The first, beta, determined the weight we gave to the query likelihood versus the answer model scores:

$$\text{ranking score} = (1-\beta)\ln P(q \mid a) + \beta \ln P(a) \tag{3}$$

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
ACM 1-58113-881-4/04/0007.

The second tuning parameter controlled the number of highest scoring bigrams in a candidate passage used to compute its perplexity.

3. EXPERIMENTAL SETUP

The AQUAINT Corpus of English News Text was used for our experiments. It consists of approximately one million documents in about 3Gb of text coming from the AP newswire, the New York Times newswire and the English portion of the Xinhua News Agency newswire. The question set consists of 500 open-domain, factoid questions such as "When was the telegraph invented?". We automatically classified questions and selected those from the following five classes: date, person, geo-political entity (gpe), definition and quantity. These question classes were selected because they encompassed the bulk of the questions (385 out of 500). And the smallest class, quantity, had enough questions (45) to be useful for the ten-fold validation protocol we used for training and testing. The accuracy of the question classification algorithm is estimated at 80% [1].

We are interested in measuring the performance of an answer passage retrieval system, not an answer extraction system. We standarized on passages with a maximum length of 250 bytes. Thus, our task is akin to the TREC 2003 QA passage sub-task.

The performance metric typically used for QA systems is the Mean Reciprocal Rank (MRR). Its use on the TREC QA tracks has varied. TREC 9 used MRR measured using the top 5 passages retrieved. We use the notation MRR(5) to denote this measure. The current measure used in the passage sub-task of the TREC QA track is rank-1 correct or in our notation, MRR(1). We report our results using these two measures and, in addition, MRR measured using the top 20 passages or MRR(20).

Each of the question classes was divided into ten partitions where 10% of the questions were held out for testing while the remaining 90% where used to obtain correct answer text to train the answer models. We wanted to use all our questions at least once for testing so the aggregate of the testing questions from all ten partitions results in the original question set. Furthermore, we wanted to avoid training answer models on text from documents that also answered test questions. The partitioning was done under the constraint that questions having answer documents in common were never split between testing and training. For example, the DATE class contained a group of five questions that share answer documents. Even though, they were not variants of each other (as in the TREC 9 QA set). This five question set is present in one of the partition's test questions and appears in the other 9 partitions as part of the training questions.

4. EXPERIMENTAL RESULTS

4.1 Query Likelihood Baseline

Our baseline came from using the query likelihood algorithm for ranking answer passages. We summarize the results in Table 1. All the performance numbers quoted in this paper correspond to the best tuning measurements. We take as our baseline the performance on the 385 questions tuned by question class since this shows a slight improvement over the same set tuned together.

Table 1: Query likelihood performance (in percentage)

Question set	MRR(1)	MRR(5)	MRR(20)
500 tuned together	25.6	31.1	32.6
385 tuned together	24.4	31.1	32.6
385 tuned apart	26.2	32.3	33.7

Table 2: Answer model performance (query likelihood baseline in parentheses)

Question class	MRR(1)	MRR(5)	MRR(20)
date	32.7(20.9)	40.5(28.5)	42.1(30.1)
person	37.4(34.1)	43.8(39.9)	44.7(40.9)
gpe	38.6(34.1)	47.6(41.2)	48.4(42.7)
definition	21.6(19.6)	25.0(22.4)	26.2(24.2)
quantity	26.7(15.6)	30.6(19.8)	31.8(21.2)
all	33.0(26.2)	39.7(32.3)	40.8(33.7)

4.2 Answer Models

Table 2 shows the performance of our bigram models trained on correct answers that have been tagged for named entities. We include the query likelihood performance in parentheses. The 'all' row refers to the combined score of the 385 questions.

5. CONCLUSIONS

Two of the five classes, 'date' and 'quantity', account for most of the improvement. The category that is most surprising is 'person'. We had expected that the presence of PERSON tokens in a passage would be a good indicator of correct answers, and thus would contribute to better performance. On the other hand, PERSON tokens are very common in news articles.

Nonetheless, the answer models improve performance under all measures. A Wilcoxon matched-pairs signed-ranks test shows that only the 'date' results are statistically significant. In the future, we plan to investigate alternative ways to abstract the training answer text used to construct the answer models.

6. REFERENCES

[1] D. Metzler and W. Croft. Analysis of statistical question classification for fact-based questions. 2003.

[2] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR 1998*, pages 275–281, 1998.

[3] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002) Conference*, 2002.

[4] M. Sabboutin and S. Sabboutin. Patterns of potential answer expressions as clues to the right answer. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, NIST Special Publication 500-250, page 293. NIST, 2001.