Simple Translation Models for Sentence Retrieval in Factoid Question Answering

Vanessa Murdock
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts
Amherst, MA 01003
vanessa@cs.umass.edu

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

ABSTRACT

Many question-answering systems start with a passage retrieval system to facilitate the answer extraction process. The richer the set of passages, in terms of answer content, the more accurate the answer extraction. We present a simple translation model for passage retrieval at the sentence level. We demonstrate this framework on TREC data, and show that it performs better than retrieval based on query likelihood, and on par with other systems.

1. INTRODUCTION

An answer can be thought of as a transformation of a question. We assume users submit queries to search engines that are well-formed questions. Some sentences will contain answers in the most direct form, whereas other sentences will contain more indirect answers. For example, given the question "Where is Glasgow?", one possible direct answer is "Glasgow is in Scotland." More typically the answer appears in context in a sentence such as, "The recession came late to Glasgow, as it did to the rest of Scotland." Both answers contain language similar to the question and can be viewed as either straightforward or more complex transformations (translations) of the question language.

Translation models, such as those described in Brown et al, [5] and [6], have been used to address the problem of machine translation. Further, they have been used in describing the language model approach to information retrieval by Berger and Lafferty [4], and in cross-lingual information retrieval by Xu, Weischedel and Nguyen [11]. We borrow these techniques in the spirit of Berger and Lafferty, for use in sentence retrieval for question answering.

Language modeling techniques rely on an accurate estimation of the probability densities of the words, given the documents. Query likelihood is an example of one language modeling technique. In the query likelihood approach, the words in a document are assumed to have a multinomial distribution. Documents are ranked according to the probability that the query was generated by random sampling from the document. Documents are sufficiently long to be distinguishable in terms of their word probability distributions. Sentences, which have considerably fewer words than documents, may be too short to accurately estimate the probability distributions of the words, or to compare those distributions to each other. We approach this problem by using the translation model to judge similarity in a slightly different way: we increase the number of direct matches between words in the question and words in the answer. The translation model allows us to match corresponding content terms that mean the same thing, or are related to each other, but that are not the same term.

A more sophisticated alignment might allow us to match answer sentences with questions more accurately, but the data doesn't lend itself to alignment in the same way that actual bilingual translation data does. There is a greater disparity in the length of the question-answer pairs. There is often no direct grammatical correlation (as in the example above). However, the translation model is good at solving the problem of synonymy. It counts matching words (such as "height" and "elevation") that query-likelihood does not count.

We demonstrate a query likelihood baseline on the TREC 2003 QA Passages task data, and show that the simplest of translation models significantly improves the retrieval of sentences. The next four sections are as follows: section 2 presents similar related work, section 3 describes the mathematical framework, section 4 presents the experiments and results, section 5 is a discussion of the work, and finally section 6 is the conclusion and future work.

2. RELATED WORK

There is a large body of work devoted to statistical approaches to language modeling, and use of translation models in cross-lingual document retrieval. The work most similar to ours use statistical translation models in a monolingual context.

Berger and Lafferty first proposed the use of translation models for (mono-lingual) document retrieval in [4]. This work describes using the IBM Model 1 to rank documents according to their "translation" probability, given the query. Model 1 assumes all possible alignments between the source sentence and the target sentence are equally likely [6]. They construct synthetic training data, in the absence of a parallel corpus of queries and relevant documents for training. They demonstrate significant improvements over their tf.idf baseline for the task of document retrieval.

For question answering, Berger et al [3] propose the use of the IBM Model 1 [5] to rank answers given questions. Their data was from the UseNet FAQs and Ben and Jerry's call center data. In this data, there is one answer for every question. They evaluated the system on five runs of randomly selected test sets of 10% of the data. It is important to note that the data was domain specific. Statistical translation performs better in domain specific applications than in general-purpose translation. For example machine translation for multilingual conference registration or airline ticketing is much more reliable than machine translation in general. Thus, these models are likely to perform better on call-center answer finding than general question answering. Still, the translation model performs significantly better than the tf.idf baseline on the same data.

Echihabi and Marcu [7] use IBM Model 4 to good effect for the problem of question answering. Question answering is the problem of finding the exact token or phrase that answers the question, separating the wheat from the chaff of the surrounding passage. They use an algorithm described in [8] to rewrite the questions as Web queries. They retrieve the top N documents from the Web, sentence segment them, rerank the sentences, and then choose the top 300 as input to the translation system. IBM Model 4 aligns parse trees in the source and the target. Parse trees show a grammatical derivation of a sentence. The leaves of the tree are the actual word tokens of the sentence. From the top 300 sentences, they use a sophisticated algorithm to construct a cut in the parse tree to select which parts of the sentence to represent by their parts of speech, and which parts to leave as the original word tokens. They use the translation model to align question and answer parse trees, and exploit the information given by the alignments and the cut to find the actual answer token. Their training data consists of the TREC 9 and 10 questions. From the documents judged relevant by TREC, they select the sentences that contain the answer patterns. They augmented this with data from the Web. The test set consists of the questions from TREC 2002, and potential answer sentences from the Web. They show a performance comparable to other question answering systems.

The University of Massachusetts submission to the TREC 2003 passage retrieval task [1] used query likelihood to rank 250-byte passages. The passages were constructed from overlapping windows of text, and then the final results were vetted to allow only one passage per document returned. The UMass system was a simple implementation, which did not take advantage of heuristically developed modules, or outside resources such as WordNet or the Web. It showed respectable results, getting a precision at rank one of around .20.

Our work is most similar to the Berger and Lafferty work in document retrieval using statistical machine translation. We use only the TREC data, and IBM translation model 1. We show that for sentence retrieval, we can improve on the query likelihood baseline without the use of the Web, or external lexicons or ontologies such as WordNet. We show that translation models provide a solid platform for sentence-based passage retrieval.

3. THE FRAMEWORK

Berger and Lafferty use the Model 1 framework to compute the probability of a query given a document. In our case, we compute the probability of a question given an answer. Since IBM Model 1 considers all alignments equally likely, it computes the probability of the question given an answer as:

$$p(Q|A) = \frac{p(m|A)}{(n+1)^m} \sum_{z_1=0}^{n} \sum_{z_2=0}^{n} \cdots \sum_{z_m=0}^{n} \prod_{i=1}^{m} p(q_i|a_{z_q}) \quad (1)$$

where m is the number of terms in the question, and n is the number of terms in the answer and $\sum_{z_i=0}^{n}$ is the sum over all possible, equally likely, alignments of a word in the question, q, to a word in the document a_{z_q}. This equation can be rewritten as:

$$p(Q|A) = \frac{p(m|A)}{(n+1)^m} \prod_{i=1}^{m} \left(\frac{n}{n+1} p(q_i|A) + \frac{1}{n+1} p(q_i|\epsilon) \right) \quad (2)$$

where ϵ is the empty word and

$$p(q_i|A) = \sum_{j=1}^{n} p(q_i|a_j) p(a_j|A) \quad (3)$$

$p(q_i|a)$ is the probability that the ith term of the question, q_i, translates to the jth term in the answer, a_j, and $p(a_j|A)$ is the probability that term a_j was generated by answer sentence A.

Since we are not generating answers or questions, rather we are ranking existing answers for a single question, the length of the question is the same for every answer sentence, and the coefficient at the beginning doesn't affect the ranking. Furthermore, since both the question and the answer are in the same vocabulary, every word has a probability of translating to itself (if nothing else), there is no possibility that a question word translates to the empty string (as might be the case if we were considering two different languages, where tokens in one language have no translation in the other). Thus, rather than compute the probability that a word translates to the empty word, we consider the probability that it is generated by some other sentence, and we estimate it from the collection:

$$p(Q|A) = \prod_{i=1}^{m} \lambda \left(\sum_{j=1}^{n} p(q_i|a_j) p(a_j|A) \right) + (1-\lambda) p(q_i|C) \quad (4)$$

where $p(q_i|C)$ is the probability that the term q_i was generated by the collection, C. Note that if term q_i translates only to itself (with probability 1) then we have exactly the formula for query likelihood:

$$p(Q|A) = \prod_{i=1}^{m} \lambda p(q_i|A) + (1-\lambda) p(q_i|C) \quad (5)$$

A major difference between machine translation and passage retrieval for QA is that machine translation assumes there is little, if any, overlap in the vocabularies of the two languages. In passage retrieval we depend heavily on the overlap between the two vocabularies. With the above formulation, the probability of a word translating to itself is estimated as a fraction of the probability of the word translating to all other words. We would hope that the model

would learn a high probability for a word translating to itself. Because the probabilities must sum to one, if there are any other translations for a given word, its self-translation probability will be less than 1.0.

To accomodate this QA-specific condition we separate out the case where a word in the question has a direct match in the answer.

Let $t_i = 1$ if $q_i = a_j$, and 0 otherwise:

$$p(q_i|a_j)p(a_j|A) = t_i p(q_i|A) + (1 - t_i) \sum_{1 \le j \le n, a_j \neq q_i} p(q_i|a_j)p(a_j|A) \quad (6)$$

Finally, since the sentences have so few terms, many of the terms in the question will not appear in the sentence. To get a better estimate of the probabilities of unseen words, we smooth with the document, as well as with the collection. The document probabilities are computed using query likelihood:

$$p(Q|D) = \prod_{i=1}^{m} \lambda p(q_i|D) + (1 - \lambda)p(q_i|C) \quad (7)$$

Since for each question there is exactly one collection, from which we are estimating the answer sentence probability, and the document probability, we combine the two scores ($p(Q|A)$ and $p(Q|D)$) with linear interpolation:

$$p(Q|A) = \prod_{i=1}^{m} \Big[\beta \Big(\lambda \sum_{j=1}^{n} p(q_i|a_j)p(a_j|A) + (1 - \lambda)p(q_i|C) \Big) + (1 - \beta) \Big(\lambda p(q_i|D_A) + (1 - \lambda)p(q_i|C) \Big) \Big] \quad (8)$$

where D_A is the document the answer sentence originated from, and C is the same collection for both answer sentences and documents. For sake of simplicity, we let the smoothing constant λ have the same value for both the document and the answer sentence.

4. EXPERIMENTS AND EVALUATION

We used as training data the TREC questions 1 through 1893. We constructed a parallel corpus by using the TREC answer patterns and judgement sets provided by NIST [1]. We selected the documents from the list of documents judged relevant for each question. From those documents, we selected the sentences that contained the answer pattern for that question. Thus the training data consisted of sentences from documents in the TREC9 corpus that contained the answer patterns for a given question. The sentences were stopped to remove single characters, and stemmed using the Krovetz stemmer. The training sentences were lower-cased, and punctuation was removed. We used GIZA [2] to learn the translation probabilities.

For the test set we used TREC questions 1894 - 2393 from the passage retrieval task, thus there were 413 test questions.

[1] http://trec.nist.gov/data.html

	MRR top 5	MRR top 20	MRR	Prec. at 1	MAP	Recall
QL	.183	.198	.203	.131	.076	.535
QL + DS	.206	.223	.228	.150	.086	.571
Model 1	.211	.222	.227	.160	.084	.560
Model 1 + DS	**.227**	**.239**†	**.244**†	**.167**†	.085	.680

Table 1: Strict assessment of TREC questions 1894 through 2393. Results that are statistically significant at a p-value < .05 are in bold, using a paired t-test. Query likelihood (first row) is the baseline. The second row is query likelihood with document smoothing. The third and fourth rows show Model 1 without and with document smoothing. "MAP" stands for mean average precision.

	MRR top 5	MRR top 20	MRR	Prec. at 1	MAP	Recall
QL	.227	.246	.251	.165	.030	.302
QL + DS	.253	.272	.277	**.184**⋆	.035	.316
Model 1	**.265**†	**.278**⋆	**.283**⋆	**.203**⋆	.032	.357
Model 1 + DS	**.275**	**.290**†	**.295**‡	**.206**†	.033	.374

Table 2: Lenient assessment of TREC questions 1894 through 2393. Results that are statistically significant at a p-value < .05 are in bold, using a paired t-test. Query likelihood (first row) is the baseline.

We retrieved the top 1000 documents for each question from the Aquaint corpus, and then sentence segmented each document. We built a separate index from the returned documents (sentence segmented) for each question. As a second pass, we retrieved the top 5000 sentences from the sentence indexes for each question. The data (questions, sentences and documents) were normalized using the Krovetz stemmer and stopped to remove single characters. As part of the indexing process, all terms in the questions and the answer were lower-cased, and punctuation was removed.

We evaluated query likelihood, with and without document smoothing, and Model 1 with and without document smoothing, using mean reciprocal rank, precision at rank one, mean average precision and recall. We did two separate evaluations, with strict and lenient criteria. For an answer to be judged correct with the strict criteria, the answer token must be in the sentence, and the sentence must be from a document listed as relevant in the TREC judgements. Under the lenient criteria, the answer must appear in the sentence. All sentences under consideration were contained in documents retrieved for a single question. Table 1 shows the results of the strict evaluation. Table 2 shows the results of the lenient evaluation. Because the TREC judgements are by no means comprehensive, the strict assessment is a lower bound on the performance of the system. Analogously, the lenient assessment is a realistic upper bound on the performance of the system.

For the strict assessment, there were a total of 9344 answers for all questions. In the lenient version there were a total of 80631 answers for all questions. The answer sentences were

Institution	Accuracy
Language Computer Corp	.685
National University of Singapore	.419
University of Waterloo	.351
University of Massachusetts	.201
Macquarie University	.191
Saarland University	.169
IIT Bombay	.133
CL Research	.119
University of Amsterdam	.111
Queens College, CUNY	.097
University of Michigan	.085

Table 3: Results of the top 10 systems on the passages task at TREC 2003.

an average of 18 tokens. The questions were an average of 7 tokens. For each question, the top 1000 documents yielded an average of 45,000 sentences.

In both table 1 and table 2, results that are statistically significant are in bold. We used a one-tailed paired t-test to evaluate the significance of the results. Results that were significant at the .05 level are in bold. Results significant at the .025 level are marked with a †. Results that were significant at the .01 level are marked with a ⋆. Results significant at the .005 level are marked with a ‡.

The MRR top 5 (and MRR top 20) are computed by averaging the inverse of the highest ranked correct answer if that answer appears in the top 5 (or top 20). The MRR is computed by taking the average of the inverse of the highest ranked correct answer over all 5000 returned sentences. The precision at one is computed by averaging the number of questions whose answer at rank one was correct.

5. DISCUSSION

TREC 2003 introduced the passage retrieval task in recognition that it is an important pre-processing step in question answering systems. Table 3 shows the results of the passage retrieval task for TREC 2003 [10]. Our result using translation models puts us in the middle of the top 10 systems, with a much simpler approach. The benefit of such a simple statistical model is that it is easy to incorporate any of the variety of other resources known to boost QA performance.

The University of Massachusetts submission to the TREC 2003 Passage Retrieval task [1] used query likelihood. In that system, passages were 250-bytes long, (roughly 45 word tokens), and they were constructed using overlapping windows of text, and then only one passage per document was selected. This system produced a precision at rank one of around .20. Having a larger amount of text to estimate the term probabilities may be an advantage. The difference between the score of the UMass system, and our query likelihood baseline of may also be attributed to the difference in the scoring process. We chose sentences because they are the intuitive unit of text for translation models. Sentences also are the intuitive input to a natural language processing system, which might make use of the structure of the sentence to extract the answer.

The translation model assigns a probability to every pair of terms that co-occur in a question and answer sentence pair. It assigns probabilities to pairs of words that don't appear often together in the question-answer pairs. Consequently, the probability mass for any given question term is spread thin if the question happens to have a lot of answers. (Not every question is guaranteed to have the same number of answers, because it is possible for a question to return fewer than 1000 documents, and each document might be a different length. Furthermore, depending on how homogeneous the results for a given question are, the size of the vocabulary of the answer candidates might vary.) We found that the model performed better if we only used translation probabilities greater than .01. All other translations are treated as OOV terms (and their probabilities estimated according to the collection).

Smoothing from the document in addition to the collection was beneficial. This allowed us to better estimate probabilities for words in relevant documents that do not appear in a given sentence. This in turn allowed us to distinguish better sentences by giving more weight to those sentences that came from better documents.

In document retrieval, there is a mismatch between the length of the query and the length of the document. It has been shown that expanding the query (for example, as in relevance models [9]) remediates this effect by making the probability distribution of the words in the query more comparable to the probability distribution of the words in the document. Query expansion may not perform as well when the document is very short, as with sentences. Translation models offer an alternative solution to this problem. Translation models allow us to count words that correspond to each other between questions and their answers, without requiring that they be the same word. Rather than expand the distribution to make the estimation of the term probabilities more accurate, the degree of overlap between the existing probability distributions is increased.

If the answer sentences in our test data were actual answers to questions, aligning the question and the answer would be more intuitive. Since a variety of other information retrieval tasks have been shown to benefit from simple bag of words models, it seems to follow that if we can estimate the words properly, then sentence retrieval could benefit from simple models as well.

6. CONCLUSION AND FUTURE WORK

Simple translation models perform well for the sentence retrieval task. Using a translation model with the most basic of alignments shows a significant improvement over the query likelihood baseline result. The retrieval of sentences is further improved by smoothing with the document probabilities, rather than just the collection probabilities.

For future work, we plan to investigate the use of relevance models, and cross-lingual relevance models for this task. We also plan to investigate the effect of the passage size on retrieval. Also left to future work is the investigation of the various translation alignment algorithms. It may be that a slightly more sophisticated alignment improves the estimation of the term probabilities.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge Andres Corrada-Emmanuel for useful discussions about the University of Massachusetts submission to the TREC 2003 QA-Passages task. We would also like to thank David Fisher for his valuable help with the Lemur Toolkit. This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-9907018, and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] N. AbdulJaleel, A. Corrada-Emmanuel, Q. Li, X. Liu, C. Wade, and J. Allan. Umass at trec 2003: Hard and QA. In *Proceedings of the Twelfth Text Retrieval Conference (TREC)*, 2003.

[2] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, I. D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. Statistical machine translation, final report, JHU workshop, 1999.

[3] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 192–199, 2000.

[4] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)*, 1999.

[5] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelineck, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[6] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263 – 311, 1993.

[7] A. Echihabi and D. Marcu. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003.

[8] U. Hermjakob, A. Echihabi, and D. Marcu. Natural language based reformulation resource and web exploitation for question answering. In *Proceedings of the TREC-2002 Conference, NIST.*, 2001. Gaithersburg, MD.

[9] V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *Proceedings of the 25th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)*, 2002.

[10] E. M. Voorhees. Overview of the trec 2003 question answering track. In *Proceedings of the Twelfth Text Retrieval Conference (TREC)*, 2003.

[11] J. Xu, R. Weischedel, and C. Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference*, 2001.