A Language Modeling Approach to Passage Question Answering

Dell Zhang[1,2]
[1] Department of Computer Science
School of Computing
S15-05-24, 3 Science Drive 2
National University of Singapore
Singapore 117543
[2] Singapore-MIT Alliance
E4-04-10, 4 Engineering Drive 3
Singapore 117576
+65-68744251
dell.z@ieee.org

Wee Sun Lee[1,2]
[1] Department of Computer Science
School of Computing
S15-05-24, 3 Science Drive 2
National University of Singapore
Singapore 117543
[2] Singapore-MIT Alliance
E4-04-10, 4 Engineering Drive 3
Singapore 117576
+65-68744526
leews@comp.nus.edu.sg

Abstract

This paper reports our efforts on developing a language modeling approach to passage question answering. In particular, we address the following two problems: (i) generalized language modeling for question classification; (ii) constrained language modeling for passage retrieval.

1 Introduction

The Text Retrieval Conference (TREC) has a Question Answering (QA) track to support large-scale evaluation for open-domain QA systems [1-4]. The TREC2003 QA track consists of two separate tasks, the main task and the passage task. We only participated in the passage task.

The passage task of a QA system is to find a small chunk of text that contains the exact-phrase answer of a given question from a large document collection. Lin et al. [5] have showed that users prefer passages over exact-phrase answers in a real-world setting because paragraph-sized chunks provide context. Furthermore, exact-phrase answers are too short to make good training data for future research, making passages a better resource.

This paper reports our efforts on developing a language modeling approach to passage question answering. In particular, we address the following two problems: (i) generalized language modeling for question classification; (ii) constrained language modeling for passage retrieval.

The rest of this paper is organized as follows. In §2, we give a brief review of the language modeling technique. In §3, we describe the architecture of our TREC2003 QA system. In §4, we describe the question classification module. In §5, we describe the passage retrieval module. In §6, we present the evaluation results. In §7, we make concluding remarks.

2 Language Modeling

The language modeling technique is originally motivated by speech recognition, and it has become widely used in many other application areas such as document classification and information retrieval. This section gives a brief review of the language modeling technique. Please be referred to [6, 7] for more detailed explanation.

The goal of language modeling, in general, is to build a language model M_L that captures the statistical regularities of natural language L. Given a word string $S = w_1 w_2 ... w_l$, M_L attempts to predict $\Pr[S \mid M_L]$ $= \Pr_L[S]$, the occurring probability of S in L.

The most common language model is the n-gram model. Despite of its simplicity, the n-gram model works quite well in practice. Applying the chain rule of probability, we get

$$\Pr_L[S] = \Pr_L[w_1 w_2 ... w_l] = \prod_{i=1}^{l} \Pr_L[w_i \mid w_1 ... w_{i-1}].$$

The n-gram model approximates this probability by assuming that the occurrence of w_i only depends on its preceding $n-1$ words, i.e.,

$$\Pr_L[w_i \mid w_1 ... w_{i-1}] = \Pr_L[w_i \mid w_{i-n+1} ... w_{i-1}].$$

A straightforward way to estimate $\Pr_L[w_i \mid w_{i-n+1} ... w_{i-1}]$ is to use maximum likelihood estimation given by

$$\Pr_L[w_i \mid w_{i-n+1} ... w_{i-1}] = \frac{\#_L(w_{i-n+1} ... w_i)}{\#_L(w_{i-n+1} ... w_{i-1})},$$

where $\#(S)$ denotes the number of occurrences of S in the training data of L. However, maximum likelihood estimation assigns zero probabilities to the n-gram strings that were never witnessed in the training data, which are obviously untrue and cause serious problems. Therefore smoothing methods should be used to adjust maximum likelihood estimation to produce more accurate probabilities. One simple but effective smoothing method is to combine the raw model M_{La} (e.g. bigram model) with its background model M_{Lb} (e.g., unigram model) by linear interpolation:

$$\Pr_L[S] = \lambda \Pr_{La}[S] + (1-\lambda) \Pr_{Lb}[S],$$

where $0 \le \lambda \le 1$ is a weighting parameter. More powerful smoothing methods include additive smoothing (e.g. Laplace smoothing), Jelinek-Mercer smoothing, Katz smoothing, Witten-Bell smoothing, Kneser-Ney smoothing, and so on [8].

3 System Overview

The architecture of our TREC2003 QA system is shown in Figure 1. It consists of two major modules: question classification and passage retrieval.

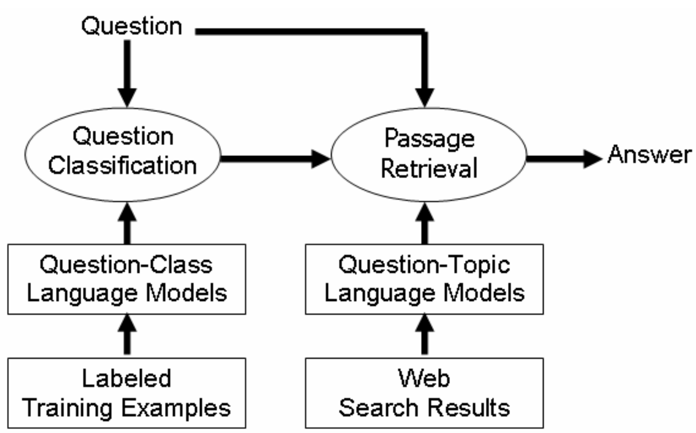

Figure 1. The architecture of our TREC 2003 QA system.

The question classification module identifies each question's preferred answer type using question-class language models, which are learned from thousands of labeled training examples. The language modeling based classification algorithm has many advantages over the popular Naive Bayes algorithm. To tackle the scarcity of training data, we build question-topic language models on generalized question structures but not specific word sequences. The generalized question structures are derived from the original questions through various lexical, syntactic and semantic generalization rules.

The passage retrieval module identifies each question's expected answer context using question-topic language models, which are learned from Web search results. Given a question, we first get a set of relevant passages from the local document collection. Then we search the Web, build a question-topic language model and augment it with a set of probabilistic constraints. Next we rank the retrieved passages using the question-topic language model. Finally, we return the highest ranked passage whose score is above a threshold as the answer. The language modeling based retrieval algorithm implicitly has the power of massive query expansion, which is helpful to overcome the lexical chasm between questions and answers.

4 Question Classification

The task of question classification could be automatically accomplished using machine learning methods [9-11]. Here we attempt to apply language modeling to question classification.

Given a question $Q = q_1 q_2 ... q_k$, it is natural to assign it to the question class which has highest posterior probability, i.e.,

$$C^* = \arg\max_C \Pr[C \mid Q].$$

The posterior probability $\Pr[C \mid Q]$ can be computed via Bayes's rule:

$$\Pr[C \mid Q] = \frac{\Pr[Q \mid C]\Pr[C]}{\Pr[Q]} \propto \Pr[Q \mid C]\Pr[C].$$

The prior probability $\Pr[C]$ can be estimated by the fraction of training questions labeled C. To estimate the probability $\Pr[Q \mid C]$, we build a question-class language model M_C for C and then get

$$\Pr[Q \mid C] = \Pr[Q \mid M_C] = \Pr_C[Q] = \Pr_C[q_1 q_2 ... q_k].$$

In our QA system, smoothed bigram models (see §2) are used to implement question-class language models.

The language modeling based classification (LMC) algorithm is very similar to the popular Naïve Bayes (NB) algorithm [12]. In fact, the LMC algorithm is a straightforward generalization of the NB algorithm: a uniram classifier with Laplace smoothing corresponds exactly to the traditional NB classifier. However, the LMC algorithm possesses many advantages over the NB algorithm, including modeling longer context with larger n and applying superior smoothing techniques in the presence of sparse data [13].

Note that the power of language modeling is often hurt by the scarcity of training data. Applying language modeling to question classification is no exception. To overcome this obstacle, we build question-topic language models on generalized question structures but not specific word sequences. For instance, a question in the form "When was *sb.* born?" always asks for a date no matter who "*sb.*" is, so if we have a DATE-class language model that can accurately predict the probability of the generalized question structure "When was <PERSON> born?", we are able to ensure correct classification of the question "When was Albert Einstein born?" even though "Albert Einstein" has never occurred in the training data.

The generalized question structures are derived from the original questions through various generalization rules, which may include:

- lexical generalization, e.g., replacing every acronym with <ACRONYM>, replacing every number with <NUMBER>;
- syntactical generalization, e.g., replacing every quoted-string with <QUOTED>, replacing every clause with <CLAUSE>;
- semantic generalization, e.g., replacing every string that is a named entity (like organization) with a tag representing its type (like <ORGANIZATION>), replacing every word that belongs to a specific semantic category (like animal) with a tag representing its hypernym (like <ANIMAL>).

The named entity recognizer is modified from a component of GATE [14] (available at http://gate.ac.uk/), and the semantic categories are defined taking advantage of WordNet (available at http://www.cogsci.princeton.edu/~wn/).

5 Passage Retrieval

Recently the language modeling technique has been introduced to information retrieval area and shown considerable success in many applications [15-19]. Here we attempt to apply language modeling to passage retrieval in QA scenario.

Given a question $Q = q_1 q_2 ... q_k$, we first get a set of relevant passages from the local document collection, using the MG software [20] (available at http://www.cs.mu.oz.au/mg/). The passages are defined as half-overlapped text windows each consisting of a fixed number (30 in our case) of words. Every passage is restricted not to cross paragraph boundary. Please be referred to [21] for a recent survey of various kinds of passages.

These passages need to be ranked according to their possibilities of containing the right answer. From the language modeling standpoint, effective ranking of passages could be achieved by constructing a question-topic language model, which represents our expectations about the answer context. The primary difficulty here is the lack of training data.

Lavrenko and Croft [15] have proposed a wise method called "relevance-based language modeling", that can build a unigram model M_R describing a topic in absence of training data. Their method is to approximate $\Pr[w \mid M_R]$ by the formula:

$$\Pr[w \mid M_R] \approx \Pr[w \mid Q] = \frac{\Pr[w, q_1, q_2, ..., q_k]}{\Pr[q_1, q_2, ..., q_k]} = \frac{\Pr[w, q_1, q_2, ..., q_k]}{\sum_w \Pr[w, q_1, q_2, ..., q_k]}.$$

To estimate the joint probability $\Pr[w, q_1, q_2, ..., q_k]$, we assume that there exists a set $\mathcal{M}$ of underlying source distributions from which w and $q_1, q_2, ..., q_k$ could have been sampled independently, then we get

$$\Pr[w, q_1, q_2, ..., q_k] = \sum_{M_D \in \mathcal{M}} \Pr[M_D] \left(\Pr[w \mid M_D] \prod_{i=1}^{k} \Pr[q_i \mid M_D] \right).$$

Thus the probability $\Pr[w \mid M_R]$ can be computed as

$$\Pr[w \mid M_R] = \sum_{M_D \in \mathcal{M}} \Pr[w \mid M_D] \Pr[M_D \mid q_1, q_2, ..., q_k].$$

Now it becomes obvious that M_R is a linear mixture of distributions from $\mathcal{M}$, where each distribution M_D is "weighted" by its posterior probability of generating the question, $\Pr[M_D \mid q_1, q_2, ..., q_k]$.

Since previous research work has revealed immense benefits of exploiting the Web data for QA [22, 23], we decide to construct $\mathcal{M}$ from the question's relevant Web search results. As in [23], we formulate several queries by rewriting the question Q, and submit these queries to a search engine like Google (http://www.google.com) to get search results. For each search result D, we build a smoothed unigram model (see §2) that is to be used as a source distribution $M_D \in \mathcal{M}$, so that $\Pr[w \mid M_D] = \Pr_D[w]$. To make the computation of $\Pr[w \mid M_R]$ tractable, we only use the top-N search results. This simplification is reasonable because the probability $\Pr[M \mid q_1, q_2, ..., q_k]$ should have near-zero values for all but the top-N search results. In practice, the strict probabilistic interpretation of $\Pr[M_D \mid q_1, q_2, ..., q_k]$ could be relaxed and substituted by any heuristic estimate, as long as it is non-negative and sums to 1 [16]. In our QA system, $\Pr[M_D \mid q_1, q_2, ..., q_k]$ is substituted by a weight of M_D whose value is set according to the precision of its corresponding query [23]. For example, the search results returned by the query "+the Louvre Museum +is located" would be weighted higher than those returned by the query "Louvre".

Furthermore, we augment the question-topic language model M_R with a set of constraints which are expressed as probabilities of various events. The constraints used in our QA system include:

- answer-type constraints, e.g., $\Pr[\bar{\mathrm{A}} \mid M_R] = 0$ that means M_R should give zero probability to passages containing no named entity of the desired answer type A;
- answer-context constraints, e.g., for a question in the form "How did sb. die?", we could force $\Pr[\text{survive} \mid M_R] = 0.0$, $\Pr[\text{wreck} \mid M_R] = 0.1$, $\Pr[\text{kill} \mid M_R] = 0.2$, $\Pr[\text{suicide} \mid M_R] = 0.2$, etc.; or we could interpolate M_R with a pre-built model $M_{die-reason}$ which is learned from question-answer pair examples on this topic.

After augmenting these constraints, M_R is adjusted to meet the requirement $\sum_w \Pr[w \mid M_R] = 1$. In this way, we are able to incorporate some prior knowledge into the question-topic language model.

What remains is to use the constructed question-topic language model M_R to rank relevant passages. For each passage P, we build a smoothed unigram model (see §2) M_P. As suggested in [16], we use the

Kullback-Leibler (KL) divergence between passage language model M_P and question-topic language model M_R to rank passages. The KL divergence (also known as relative entropy) between M_P and M_R is defined as:

$$divergence(M_P \| M_R) = \sum_w \Pr[w \mid M_P] \log \frac{\Pr[w \mid M_P]}{\Pr[w \mid M_R]}.$$

Passages whose language models have a smaller divergence with the question-topic language model are considered more relevant to the question's topic. The KL divergence yields a reasonable ranking metric, but has problems when straightforwardly used in QA scenario. Consider a passage P which is very vague (looks too much like general English), it is unlikely to contain the right answer even if $divergence(M_P \| M_R)$ is small, because it does not describe a specific topic. To avoid such trivial passages, we leverage a notion of language model clarity [17]. Given a passage language model M_P, its clarity is defined as $clarity(M_P) = divergence(M_P \| M_G)$, where M_G is the language model of general English estimated from a very large corpus. Consequently we rank the relevant passages according to the following score function:

$$\begin{aligned}
score(P) &= -divergence(M_P \| M_R) + clarity(M_P) \\
&= -divergence(M_P \| M_R) + divergence(M_P \| M_G) \\
&= -\sum_w \Pr[w \mid M_P] \log \frac{\Pr[w \mid M_P]}{\Pr[w \mid M_R]} + \sum_w \Pr[w \mid M_P] \log \frac{\Pr[w \mid M_P]}{\Pr[w \mid M_G]} \\
&= \sum_w \Pr[w \mid M_P] \log \frac{\Pr[w \mid M_R]}{\Pr[w \mid M_G]}.
\end{aligned}$$

That is, the degree to which M_P is similar to M_R, increased to the extent that M_P is a clear (focused) model that differs from general English. Note that adding clarity has resulted in the denominator that plays a role similar to IDF in standard information retrieval [24]. Finally, we return the highest ranked passage whose score is above a threshold as the answer. If no such answer could be found, we return 'NIL'.

Massive query expansion is an integral part of the language modeling based retrieval algorithm, because we compute the probability $\Pr[w \mid M_R]$ for every word in the language. This helps our QA system to overcome the lexical chasm between questions and answers.

6 Evaluation

The document set for evaluation is the AQUAINT collection that consists of 1,033,461 documents taken from the New York Times, the Associated Press, and the Xinhua News Agency newswires. The question set for evaluation contains 413 factoid questions that seek short, fact-based answers.

A submission for the passage task must contain exactly one answer for each factoid question. An answer is either "NIL" or an extracted passage from a document. A passage should be no longer than 250 bytes, and judged either incorrect (does not contain a correct answer), unsupported (contains a correct answer, but the document doesn't say so), or correct. Unresponsive passages (a passage that refers to an imitation or copy; a passage that contains multiple instances of the correct semantic category of the answer without actually specifying which is the answer; passages that omit necessary units; etc.) are incorrect. For a question with no correct answer in the document collection, only "NIL" answer is correct. The final score for a passage task submission is its accuracy (the fraction of answers judged correct).

The official evaluation result of our TREC2003 QA system is shown in Table 1.

#(test questions)	413
#(correct answers)	173
#(unsupported answers)	9
#(incorrect answers)	231
accuracy	173 / 413 = 0.419
precision of recognizing no answer	10 / 64 = 0.156
recall of recognizing no answer	10 / 30 = 0.333

Table 1. The evaluation result of our TREC2003 QA system.

7 Conclusion

This paper reports our efforts on developing a language modeling approach to passage question answering. We want to demonstrate and advocate that language modeling may provide a uniform framework in which QA systems can integrate evidences from multiple knowledge sources to find the right answer.

Possible future work include: extending this language modeling approach to handle definition questions and list questions; integrating textual patterns [22] into language models; building language models to exploit structured and semi-structured data, particularly HTML/XML data on the Web.

References

[1] E. M. Voorhees, "The TREC-8 Question Answering Track Report," in *Proceedings of the 8th Text Retrieval Conference (TREC)*. Gaithersburg, MD: NIST, 1999, pp. 77-82.

[2] E. M. Voorhees, "Overview of the TREC-9 Question Answering Track," in *Proceedings of the 9th Text Retrieval Conference (TREC)*. Gaithersburg, MD: NIST, 2000, pp. 71-80.

[3] E. M. Voorhees, "Overview of the TREC 2001 Question Answering Track," in *Proceedings of the 10th Text Retrieval Conference (TREC)*. Gaithersburg, MD: NIST, 2001, pp. 157-165.

[4] E. M. Voorhees, "Overview of the TREC 2002 Question Answering Track," in *Proceedings of the 11th Text Retrieval Conference (TREC)*. Gaithersburg, MD: NIST, 2002, pp. 57-68.

[5] J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz, and D. R. Karger, "The Role of Context in Question Answering Systems," presented at Conference on Human Factors and Computing Systems (CHI), Fort Lauderdale, Florida, 2003.

[6] J. Goodman, "A Bit of Progress in Language Modeling, Extended Version," Microsoft Research, Technical Report MSR-TR-2001-72, 2001.

[7] R. Rosenfeld, "Two Decades Of Statistical Language Modeling: Where Do We Go From Here?," *Proceedings of the IEEE*, vol. 88, pp. 1270-1278, 2000.

[8] S. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Science Group, Harvard University, Technical Report TR-10-98, 1998.

[9] W. Li, "Question Classification Using Language Modeling," in *CIIR Technical Report*: University of Massachusetts, Amherst, 2002.

[10] X. Li and D. Roth, "Learning Question Classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*. Taipei, Taiwan, 2002, pp. 556-562.

[11] D. Zhang and W. S. Lee, "Question Classification using Support Vector Machines," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Toronto, Canada, 2003, pp. 26- 32.

[12] T. Mitchell, *Machine Learning*, international ed. Singapore: McGraw Hill, 1997.

[13] F. Peng and D. Schuurmans, "Combining Naive Bayes and n-Gram Language Models for Text Classification," in *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR)*. Pisa, Italy, 2003, pp. 335-350.

[14] H. Cunningham, "GATE, a General Architecture for Text Engineering," *Computers and the Humanities*, vol. 36, pp. 223-254, 2002.

[15] V. Lavrenko and W. B. Croft, "Relevance-Based Language Models," in *Proceedings of 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. New Orleans, LA, 2001, pp. 120-127.

[16] V. Lavrenko, M. Choquette, and W. B. Croft, "Cross-Lingual Relevance Models," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Tampere, Finland, 2002, pp. 175-182.

[17] V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas, "Relevance Models for Topic Detection and Tracking," in *Proceedings of the Human Language Technology Conference (HLT)*. San Diego, CA, 2002, pp. 104-110.

[18] X. Liu and W. B. Croft, "Passage Retrieval based on Language Models," in *Proceedings of the 11th ACM CIKM International Conference on Information and Knowledge Management (CIKM)*. McLean, VA, 2002, pp. 375-382.

[19] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic Image Annotation and Retrieval using Cross-Media Relevance Models," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Toronto, Canada, 2003, pp. 119-126.

[20] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 1999.

[21] M. Kaszkiel and J. Zobel, "Effective Ranking with Arbitrary Passages," *Journal of the American Society for Information Science and Technology*, vol. 52, pp. 344-364, 2001.

[22] D. Zhang and W. S. Lee, "Web based Pattern Mining and Matching Approach to Question Answering," in *Proceedings of the 11th Text Retrieval Conference (TREC)*. Gaithersburg, MD: NIST, 2002, pp. 497-504.

[23] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng, "Web Question Answering: Is More Always Better?," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Tampere, Finland, 2002, pp. 291-298.

[24] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York, NY: Addison-Wesley, 1999.