TREC2006 Question Answering Experiments at Tokyo Institute of Technology

Edward Whittaker Josef Novak Pierre Chatain Sadaoki Furui
Dept. of Computer Science
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku
Tokyo 152-8552 Japan
{edw,novakj,pierre,furui}@furui.cs.titech.ac.jp

Abstract

In this paper we describe Tokyo Institute of Technology's speech group's second attempt at the TREC2006 question answering (QA) track. Keeping the same theoretical QA model as for the TREC2005 task this year we investigated combinations of variations of models focusing once again on the factoid QA task. An experimental run combining translated answers from separate English, French and Spanish systems proved inconclusive. However, our best combination of all component models gave us a factoid performance of 25.1% (placing us 9th and well above the median of the 30 participating systems of 18.6%) and an overall performance including the results from the list and other question tasks of 11.6% (which was somewhat below the median of 13.4%).

1 Introduction

In this paper, we describe the application of our data-driven and non-linguistic framework for the factoid QA task of TREC2006 that was applied succesfully in the TREC2005 Question Answering (QA) track [7]. For convenience we copy verbatim the exposition of our mathematical model for question answering in Section 2.

Three runs were submitted for evaluation (`asked06a,b,c`) that comprised various combinations of QA systems, primarily based on systems employing our novel, statistical approach. For the list task, an extension to the system used in the factoid QA task was used. For the *other* question task a variation on a system used for speech summarization [3] was used, identical to one of the systems used last year.

Description of system settings and results for this year's task are given in Section 7. A discussion and conclusion are given in Sections 8 and 9.

2 Factoid question task

This section is re-produced verbatim from the paper "TREC2005 Question Answering Experiments at Tokyo Institute of Technology" [7].

It is clear that the answer to a question depends primarily on the question itself but also on many other factors such as the person asking the question, the location of the person, what questions the person has asked before, and so on. Although such factors are clearly relevant in a real-world scenario they are difficult to model and also to test in an off-line mode, for example, in the context of the TREC evaluations. We therefore choose to consider only the dependence of an answer A on the question Q, where each is considered to be a string of l_A words $A = a_1, \ldots, a_{l_A}$ and l_Q words $Q = q_1, \ldots, q_{l_Q}$, respectively. In particular, we hypothesize that the answer A depends on two sets of features $W = \mathcal{W}(Q)$ and $X = \mathcal{X}(Q)$ as follows:

$$P(A \mid Q) = P(A \mid W, X), \tag{1}$$

where $W = w_1, \ldots, w_{l_W}$ can be thought of as a set of l_W features describing the "question-type" part of Q such as *when*, *why*, *how*, etc. and $X = x_1, \ldots, x_{l_X}$ is a set of l_X features comprising the "information-bearing" part of Q i.e. what the question is actually about and what it refers to. For example, in the questions, *Where was Tom Cruise married?* and *When was Tom Cruise married?* the information-bearing component is identical in both cases whereas the question-type component is different.

Finding the best answer $\hat{A}$ involves a search over all A for the one which maximizes the probability of the above model:

$$\hat{A} = \arg\max_A P(A \mid W, X). \tag{2}$$

This is guaranteed to give us the optimal answer in a maximum likelihood sense if the probability distribution is the correct one. We don't know this and it's still difficult to model so we make various modeling assumptions to simplify things. Using Bayes' rule this can be rearranged as

$$\arg\max_A \frac{P(W, X \mid A) \cdot P(A)}{P(W, X)}. \quad (3)$$

The denominator can be ignored since it is common to all possible answer sequences and does not change. Further, to facilitate modeling we make the assumption that X is conditionally independent of W given A to obtain:

$$\arg\max_A P(X \mid A) \cdot P(W \mid A) \cdot P(A). \quad (4)$$

Using Bayes rule, making further conditional independence assumptions and assuming uniform prior probabilities, which therefore do not affect the optimisation criterion, we obtain the final optimisation criterion:

$$\arg\max_A \underbrace{P(A \mid X)}_{\substack{retrieval \\ model}} \cdot \underbrace{P(W \mid A)}_{\substack{filter \\ model}}. \quad (5)$$

The $P(A \mid X)$ model is essentially a language model which models the probability of an answer sequence A given a set of information-bearing features X, similar to the work of [6]. It models the proximity of A to features in X. We call this model the *retrieval model* and examine it further in Section 2.1.

The $P(W \mid A)$ model matches an answer A with features in the question-type set W. Roughly speaking this model relates ways of asking a question with classes of valid answers. For example, it associates dates, or days of the week with *when*-type questions. In general, there are many valid and equiprobable A for a given W so this component can only re-rank candidate answers retrieved by the retrieval model. If the filter model were perfect and the retrieval model were to assign the correct answer a higher probability than any other answers of the same type the correct answer should always be ranked first. Conversely, if an incorrect answer, in the same class of answers as the correct answer, is assigned a higher probability by the retrieval model we cannot recover from this error. Consequently, we call it the *filter model* and examine it further in Section 2.2.

2.1 Retrieval model

The retrieval model essentially models the proximity of A to features in X. Since $A = a_1, \ldots, a_{l_A}$ we are actually modeling the distribution of multi-word sequences. This should be borne in mind in the following discussion whenever A is used. As mentioned above, we currently use a deterministic information-feature mapping function $X = \mathcal{X}(Q)$. This mapping only generates word m-tuples ($m = 1, 2, \ldots$) from single words in Q that are not present in a *stop-list* of around 50 high-frequency words. In principle the function could of course extract deeper linguistic features but we leave this for future work.

We first assume that a corpus of text data S is available for searching for answers comprising $|S|$ sentences $S_1, \ldots, S_{|S|}$ and $|U|$ documents and a vocabulary V of $|V|$ unique words. We use the notation X_i to define an active set of the features $x_1, \ldots, x_{l_X}$ such that $X_i = x_1 \cdot \delta(d_1), x_2 \cdot \delta(d_2), \ldots, x_{l_X} \cdot \delta(d_{l_X})$ where $\delta(\cdot)$ is a discrete indicator function which equals 1 if its argument evaluates true (i.e. its argument(s) are equal, is not an empty set, or is a positive number) and 0 if false (i.e. its argument(s) are not equal, is an empty set, is 0 or is a negative number) and $\vec{d} = [d_1, \ldots, d_{l_X}]$ is the solution[1] to $i = \sum_{j=1}^{l_X} 2^{j-1} d_j$.

The probability $P(A \mid X)$ is modeled as a linear interpolation of the 2^{l_X} distributions[2]:

$$P(A \mid X) = \sum_{i=0}^{2^{l_X - 1}} \lambda_{X_i} \cdot P(A \mid X_i), \quad (6)$$

where $\lambda_{X_i} = 1/2^{l_X}$ for all i, $P(A \mid X_0)$ is a zerogram distribution, and $P(A \mid X_i)$ is the conditional probability of A given the feature set X_i and is computed as the maximum likelihood estimate from the corpus S:

$$P(A \mid X_i) = \frac{N(A, X_i)}{N(X_i)}, \quad (7)$$

where

$$N(A, X_i) = \sum_{j=1}^{|S|} \delta(X_i \in \mathcal{X}(S_j)) \cdot \delta(A \in S_j), \quad (8)$$

$$N(X_i) = \sum_{v \in V} N(v, X_i). \quad (9)$$

We modify Equation (8) to include contributions from adjacent sentences weighted by λ_{adj} which typically has a value ≤ 1:

[1] Note that the value of i is simply the base10 number that represents the binary encoding of the active features in X_i.

[2] A linear interpolation of models, which borrows directly from statistical language modeling techniques for speech recognition, was found to give retrieval performance approximately twice that of a naive-Bayes or log-linear formulation.

$$N(A, X_i) = \sum_{j=1}^{|S|} \delta(X_i \in \mathcal{X}(S_j)) \cdot \max\{\delta(A \in S_j), \lambda_{adj} \cdot \delta(A \in S_{j-1}), \lambda_{adj} \cdot \delta(A \in S_{j+1})\}. \quad (10)$$

It turns out that smoothing the maximum likelihood estimates from each component distribution has little effect on performance so none is performed. This is partly because of the inherent smoothing effect achieved by interpolating all the distributions together and partly since there is no need to smooth for non-occurring events since such zerotons are never likely to be selected as answers.

One clear deficiency, however, is the use of equal-valued interpolation weights for all distributions. One might expect a dependence on the number of active features or on $N(X_i)$, however, no such reliable relationship has so far been determined although investigations continue.

2.2 Filter model

The question-type mapping function $\mathcal{W}(Q)$ extracts n-tuples ($n = 1, 2, \ldots$) of question-type features from the question Q, such as *How*, *How many* and *When were*. A set of $|V_{\mathcal{W}}| = 2522$ single-word features is extracted based on frequency of occurrence in questions in previous TREC question sets. Some examples include: *when*, *where*, *who*, *whose*, *how*, *many*, *high*, *deep*, *long* etc.

Modeling the complex relationship between W and A directly is non-trivial. We therefore introduce an intermediate variable representing classes of example questions-and-answers (q-and-a) c_e for $e = 1 \ldots |C_E|$ drawn from the set C_E, and to facilitate modeling we say that W is conditionally independent of c_e given A as follows:

$$P(W \mid A) = \sum_{e=1}^{|C_E|} P(W, c_e \mid A) \quad (11)$$

$$= \sum_{e=1}^{|C_E|} P(W \mid c_e) \cdot P(c_e \mid A). \quad (12)$$

Given a set E of example q-and-a t_j for $j = 1 \ldots |E|$ where $t_j = (q^j_1, \ldots, q^j_{l_{Q^j}}, a^j_1, \ldots, a^j_{l_{A^j}})$ we define a mapping function $f : E \mapsto C_E$ by $f(t_j) = e$. Each class $c_e = (w^e_1, \ldots, w^e_{l_{W^e}}, a^e_1, \ldots, a^e_{l_{A^e}})$ is then obtained by $c_e = \bigcup_{j:f(t_j)=e} \mathcal{W}(t_j) \bigcup_{i=1}^{l_{A^j}} a^j_i$, so that:

$$P(W \mid A) = \sum_{e=1}^{|C_E|} P(W \mid w^e_1, \ldots, w^e_{l_{W^e}}) \cdot P(a^e_1, \ldots, a^e_{l_{A^e}} \mid A). \quad (13)$$

Assuming conditional independence of the answer words in class c_e given A, and making the modeling assumption that the jth answer word a^e_j in the example class c_e is dependent only on the jth answer word in A we obtain:

$$P(W \mid A) = \sum_{e=1}^{|C_E|} P(W \mid c_e) \cdot \prod_{j=1}^{l_{A^e}} P(a^e_j \mid a_j). \quad (14)$$

Since our set of example q-and-a cannot be expected to cover all the possible answers to questions that may be asked we perform a similar operation to that above to give us the following:

$$P(W \mid A) = \sum_{e=1}^{|C_E|} P(W \mid c_e) \prod_{j=1}^{l_{A^e}} \sum_{a=1}^{|C_A|} P(a^e_j \mid c_a) P(c_a \mid a_j), \quad (15)$$

where c_a is a concrete class in the set of $|C_A|$ answer classes C_A. The independence assumption leads to underestimating the probabilities of multi-word answers so we take the geometric mean of the length of the answer (not shown in Equation (15)) and normalize $P(W \mid A)$ accordingly.

The system using the above formulation of filter model given by Equation (15) is referred to as model ONE. Systems using the model given by Equation (13) are referred to as model TWO. The training of Model ONE has been described in detail in [8].

2.3 Reconciling $P(A \mid X)$ and $P(W \mid A)$

The approach to QA that has been presented is similar in essence to that of approaches to automatic speech recognition (ASR) where there are separate acoustic and language models. In ASR, it is necessary to include a *language model weight*, α, which raises the probabilities given by the language model to the power α, otherwise performance is very poor:

$$\hat{A} = \arg\max_A \frac{P(A \mid X)^\alpha \cdot P(W \mid A)}{\sum_{A'} P(A' \mid X)^\alpha \cdot P(W \mid A')}.$$

Several, possibly related, explanations have been given for this requirement including compensation for the independence assumption. In any case, the dynamic range of the models is typically very different and needs compensating somehow. α can be optimised easily once the individual models have been optimised separately.

3 List question task

For the list task we used an identical system to last year's evaluation system which itself is very similar to those systems used in the factoid task. Our factoid QA systems always output a list of all the possible answers they encounter in the data, ranked by their probabilities. The issue for the list task is therefore to determine how many of the top answers to output so as to maximise the F-score. Having investigated different methods during the development phase last year for selecting output thresholds we instead chose simply to output the top 10 answers after system combination had been performed.

4 Other question task

As in last year's evaluation we treat the answering of *other* questions as a summarization task and employ a variation on a method used for speech summarization [3] for this purpose. This year we chose to only extract nuggets from the AQUAINT corpus (rather than also extracting them from web data) since this data source demonstrated the best results in last year's evaluation. The data for each question, from which the nuggets are to be extracted, is first cleaned to remove words that are unlikely to be required in a nugget but which occur frequently in the data. Duplicate sentences are also removed along with sentences shorter than 40 bytes and longer than 220 bytes. We then select up to 500 sentences which contain as many of the topic words associated with the question as possible, assigning a score to each topic word based on an idf value obtained from the AQUAINT corpus. This results in a single document which is then summarized by selecting up to 175 important sentences according to a combination of a linguistic score (using a 3-gram language model) and a significance score (measured by a tf/idf score), according to the following:

$$S(W) = \frac{1}{N}\sum_{i=1}^{N}\{L(w_i) + \alpha \cdot I(w_i)\}, \qquad (16)$$

where N is the number of words in the sentence W, and $L(w_i)$ and $I(w_i)$ are the linguistic score and the significance score of word w_i, respectively. Sentences over 140 bytes are compacted so that all nuggets have a length between 40 and 140 bytes, using a similar summarization process. Finally, upto NU_{max} nuggets are selected according to their final summarization score, making sure that the byte-wise Levenstein distance between two nuggets is less than R% of the bytes in any previously selected sentence. Once the set of nuggets had been determined no attempt was made to suppress nuggets that contained answers already given for factoid or list questions.

5 System combination

For all runs, this year, the answers for the factoid and list tasks were all generated through some form of system combination since this was found to give the best results both in last year's TREC2005 evaluation and also during development.

The answers for run `asked06a` were produced through the weighted combination of the output of an English, French and Spanish model ONE system where the weights had been optimised during development. Answers from the French and Spanish systems were translated in to English prior to combination and up to 100 answers from each component run are considered during the combination where the score for an answer is determined as follows:

$$score(a) = \sum_{s}\frac{1}{x_s r_s(a)}, \qquad (17)$$

where x_s refers to the weight for system s, and $r_s(a)$ is the rank of answer a in system s, given the current question. If a is not output by system s we define $r_s(a) = \infty$. The answers, sorted by their new score, then form the ranked output of the combined system.

Answer combination for runs `asked06b,c` was performed by simply summing the inverse rank of an answer a from each component system s to generate a new score for the answer as follows:

$$score(a) = \sum_{s}\frac{1}{r_s(a)}. \qquad (18)$$

For the *other* question task, no system combination was performed.

6 Support generation

As for the TREC2005 evaluation an adapted version of the projection component *ProjectAnswer* of the Aranea system [5] was used for generating answer support from the AQUAINT corpus for each of the answers found using the web. Only the (upto) 1000 documents retrieved by the PRISE search engine and provided by NIST were used for searching for support information for each question. The same tool was used for determining support for answers in all 3 tasks and all runs.

System	Data source	Which model	Languages	Submitted run
asked06a	Web	ONE	English,French,Spanish	yes
asked06b	Web	ONE+TWO	English	yes
asked06c	Web	ONE+TWO	English	yes
asked06A	Web	Aranea	English	no
asked06S	Web	ONE	Spanish	no
asked06F	Web	ONE	French	no
asked06E	Web	ONE	English	no

Table 1. Descriptions of systems developed for TREC2006 including the 3 submitted runs `asked06a,b,c` **and 4 component runs** `asked06A,S,F,E` **that were not submitted for evaluation.**

System	Factoid task			List task	Other task	Avg. per-series score
	Right	Unsupp.	ineXact			
asked06a	62 (15.4%)	12 (3.0%)	24 (6.0%)	0.052	0.064	0.085
asked06b	95 (23.6%)	22 (5.5%)	27 (6.7%)	0.074	0.062	0.116
asked06c	101 (25.1%)	26 (6.5%)	27 (6.7%)	0.057	0.060	0.116

Table 2. Performance on the 3 tasks of the 3 submitted runs.

7 Experimental work

Three different systems (`asked06a,b,c`) were submitted for evaluation with characteristics given in Table 1.

System `asked06a` uses only systems based on model ONE and using web data combining the results from English, French and Spanish mono-lingual systems where the questions are translated into the target language (as appropriate) and the answers translated back in to English before combination. System `asked06b` uses a combination of a model ONE and model TWO system and only Web data. System `asked06c` combines answers from the set of unique runs that make up systems `asked06a,b`.

7.1 Question pre-processing

Conversion from the XML format provided by NIST to that required by our system was elementary. For each question set the target is extracted and each component question extracted. All target and question strings are then mapped to upper-case. All punctuation except for "'S" is removed both from target and question strings. Then, if the target for a question does not appear character-for-character in that question string it is simply appended to the end of the question string. In general, we feel our approach is quite robust to errors in pre-processing so we do not worry too much about it.

In addition, although the questions in each set are supposed to be part of a dialogue in which subsequent questions can reference prior questions and answers in the same set, we do not attempt to exploit this. Consequently, each question is treated independently of all other questions.

7.2 Target document preparation

Our system was designed with web-based question answering in mind. The source of documents we used was obtained by passing each pre-processed, upper-cased question as-is to a web search engine; the top 500 text or HTML documents returned were then downloaded and kept separate for each question. (We relied on the web search engine to strip out stop words from the query.) In contrast to other experiments using web data in the literature [1] none of our experiments has yet found a point at which performance deteriorates after a certain number of documents. We therefore settled on 500 documents for reasons of expediency rather than optimality. Subsequent text processing of the downloaded documents proceeds in essentially the same way as for question pre-processing except that HTML markup is also removed and sentence boundaries are inserted.

7.3 Factoid question task

For system development this year we only optimised performance on the TREC2004 and TREC2005 evaluation questions. Apart from several parameter settings the system used was identical to that used in the TREC2005 evaluation. For training the filter model we use 288812 example q-and-a from the Knowledge Master KM data [2] plus 2408

q-and-a from the TREC-8,9 and TREC2001 questions, and also the TREC2002,3,4 evaluation q-and-a.

The most frequent $|V_{C_A}| = 224000$ words from the AQUAINT corpus were used to obtain C_A for $|C_A| = 50, 500, 5000$ clusters as described in [8]. The vocabulary V_{C_A} covers approximately 90% of the answers in E. The maximum number of features used in the retrieval model was set to $l_X = 15$ for reasons of speed and memory efficiency.

The results for all 3 submitted runs on all 3 tasks are shown in Table 2.

7.4 List question task

This year once again no development was performed on list questions. Using the same system for answering list questions as was used last year we simply selected the top 10 scoring answers output by each system combination.

7.5 Other question task

In TREC2005 good results were obtained on the other question task using nuggets obtained only on the AQUAINT data (rather than using web data). Consequently, we used the best performing system for other questions from TREC2005 in this year's evaluation.

8 Discussion and analysis

In this year's evaluation our focus was on the factoid task and on system combination for improved performance. In last year's evaluation and during previous development our method of system combination had been found to be robust and effective at boosting overall system performance.

This year, we also looked at how we could exploit our mono-lingual QA systems in other languages to increase the variation of documents we considered (and hopefully the orthogonality of answers that are considered for combination) as well as possibly increasing our ability to answer questions about events, persons and locations in other countries. Run `asked06a` combined our mono-lingual web QA systems for English, French and Spanish. Unfortunately this turned out to be our worst run. Inexact or incorrect translations contributed somewhat to this poor result, however, the main culprit was a simple human error that was introduced during the recombination of the multi-lingual results when modifying them to conform to the TREC submission format. This scrambled the order of the final one third of the answers for this run, and consequently lowered the overall score. Unfortunately, the error was not discovered until after the results for the run had been returned from NIST. Over the portion of answers that were not corrupted the percentage of correct and supported answers was 19.4% rather than the 15.4% that was obtained over all answers.

Run `asked06b` used a similar combination of component runs as one of last year's best runs and gave a performance on the factoid task this year of 23.6%. The inclusion in run `asked06c` of the translated French and Spanish runs and also a run from a modified version of the open-source Aranea system [5] improved system performance by 1.5% absolute to 25.1%. Most of this increase probably comes from the inclusion of the Aranea answers rather than the translated multi-lingual runs though this will be more thoroughly investigated in the future.

The performance of our best run on the factoid questions increased from 21.3% last year to 25.1% this year with only a small increase in the number of ineXact and Unsupported answers. Nonetheless the absolute values of ineXact and Unsupported answers were high as they were last year. This is not particularly surprising as the projection algorithm which generates support for the answers found on the web was identical to last year and has fairly consistently shown that we suffer a loss of around 20% of our potentially correct answers due to incorrect support information. Consequently the small increase in Unsupported answers is in line with the overall increase in correct answers.

Although last year's system and this year's were not very different, it is difficult to make any hard conclusions since absolute performance is heavily dependent on the questions asked. A subsequent investigation of the relative performance against other groups will allow some assessment of the question difficulty this year compared to last year.

Answers to list questions showed small but insignificant improvements over last year's results but other answers scored approximately half of what they did last year. This was the primary factor in our signficantly reduced per series scores this year and a change in the weighting scheme that weighted each of the three tasks equally.

9 Conclusion

In this paper we have given a preliminary overview of our work for the TREC2006 question answering evaluation. Our primary focus was on the factoid task and our best run gave performance significantly higher than the median performance of all participants and substantially higher than last year's performance in TREC2005. In this evaluation our performance in the list task was not exceptional but comparable to last year's results, due to the overly simplified modelling of list questions and selection of how many answers to output. Other questions showed a marked reduction in score compared to last year and this together with a new weighting scheme that weighted each of the three tasks equally combined to give substantially lower

per-series scores than last year.

10 Online demonstration

A demonstration of the system using model ONE supporting questions in English, Japanese, Chinese, French, Spanish, Russian and Swedish can be found online at `http://asked.jp/`

11 Acknowledgments

This research was supported by JSPS and the Japanese government 21st century COE programme.

References

[1] 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*, Tampere, Finland, 2002.

[2] A. Hallmarks. Knowledge Master Educational Software. PO Box 998, Durango, CO 81302 http://www.greatauk.com/, 2002.

[3] T. Kikuchi, S. Furui, and C. Hori. Automatic speech summarization based on sentence extraction and compaction. In *Proceedings of ICASSP*, Hong Kong, China, 2003.

[4] J. Lin and D. Demner-Fushman. Automatically Evaluating Answers to Definition Questions. Technical Report LAMP-TR-119/CS-TR-4695/UMIACS-TR-2005-04, University of Maryland, 2005.

[5] J. Lin and B. Katz. Question Answering from the Web Using Knowledge Annotation and Knowledge Mining Techniques. In *Proceedings of Twelfth International Conference on Information and Knowledge Management (CIKM 2003)*, 2003.

[6] J. Ponte and W. Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*, Melbourne, Australia, 1998.

[7] E. Whittaker, P. Chatain, S. Furui, and D. Klakow. TREC2005 Question Answering Experiments at Tokyo Institute of Technology. In *Proceedings of the 14th Text Retrieval Conference*, 2005.

[8] E. Whittaker, S. Furui, and D. Klakow. A Statistical Pattern Recognition Approach to Question Answering using Web Data. In *Proceedings of Cyberworlds*, 2005.