Question Answering Using Maximum Entropy Components

Abraham Ittycheriah, Martin Franz, Wei-Jing Zhu, Adwait Ratnaparkhi
P.O.Box 218,
Yorktown Heights, NY 10598
{abei,franzm,wjzhu,adwaitr}@watson.ibm.com

Richard J. Mammone
Dept. of Electrical Engineering, Rutgers University,
Piscataway, NJ 08854
mammone@caip.rutgers.edu

Abstract

We present a statistical question answering system developed for TREC-9 in detail. The system is an application of maximum entropy classification for question/answer type prediction and named entity marking. We describe our system for information retrieval which did document retrieval from a local encyclopedia, and then expanded the query words and finally did passage retrieval from the TREC collection. We will also discuss the answer selection algorithm which determines the best sentence given both the question and the occurrence of a phrase belonging to the answer class desired by the question. A new method of analyzing system performance via a transition matrix is shown.

1 Introduction

Systems that perform question answering automatically by computer have been around for some time as described by (Green et al., 1963). Only recently though have systems been developed to handle huge databases and a slightly richer set of questions. The types of questions that can be dealt with today are restricted to be short answer fact based questions. In TREC-8, a number of sites participated in the first question-answering evaluation (Voorhees and Tice, 1999) and the best systems identified four major subcomponents:

- Question/Answer Type Classification
- Query expansion/Information Retrieval
- Named Entity Marking
- Answer Selection

Our system architecture for this year was built around these four major components as shown in Fig. 1. Here, the question is input and classified as asking for an answer whose category is one of the named entity classes to be described below. Additionally, the question is presented to the information retrieval (IR) engine for query expansion and document retrieval. This engine, given the query, looks at the database of documents and outputs the best documents or passages annotated with the named entities. The final stage is to select the exact answer, given the information about the answer class and the top scoring passages. Minimizing various distance metrics applied over phrases or windows of text results in the best scoring section that has a phrase belonging to the answer class. This then represents the best scoring answer.

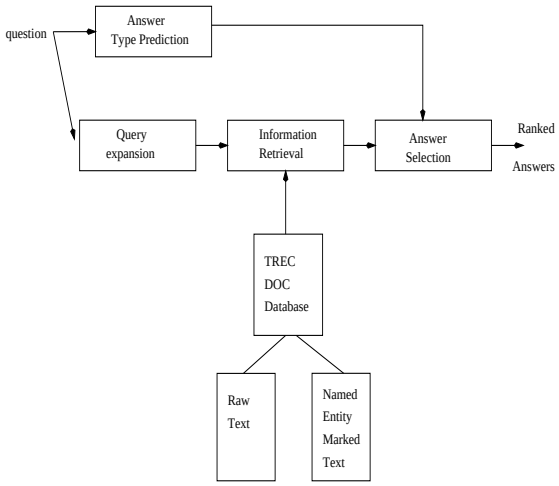

Figure 1: Question Answering Architecture

Maximum entropy modelling is described in (Della Pietra et al., 1995; Berger et al., 1996). Methods of feature selection is further described in (Berger and Printz, 1998). We will not discuss the mathematical details of the algorithm here, instead we will only show the features that

are used in such a model.

The paper is organized as follows: we will describe Answer Type Classification, Information Retrieval and Answer Selection in the next 3 sections and then analyze the system using the TREC9 dataset in the next section and conclude.

2 Answer Type Classification

In answer type classification the problem is to label a question with the label of the named entity that the question seeks. Our labels are the standard MUC (Chinchor, 1997) categories with the addition of PHRASE which is a catch all for answers not of the standard categories. These categories are

- Person, Location, Organization
- Cardinal, Percent
- Date, Time, Duration
- Measure
- Money
- Phrase, Reason

The REASON category was tied to WHY questions. Processing of REASON and PHRASE is the same in our system, interpreting it as desiring a clause which had a noun phrase embedded in it.

A corpus of questions that has been annotated with the above mentioned categories was created. We created 1900 questions by presenting a human subject a document selected at random and having read a portion of the document, a question was phrased; the answer and the document number are noted in addition. We also used 1400 questions from a trivia database (Academic Hallmarks, 1999) annotated in a similar manner. This data is used to train the maximum entropy classifier. A separate set of 182 questions is used as the heldout test set.

In the experiments, the types of feature functions shown in Table 1 were used. Each feature type expands on the one above it. The "Expanded Hierarchy" feature type uses WordNet (Miller, 1990) to expand words from a question word upto and including the first noun cluster. The "Mark Question Word" feature type identifies the question words and labels them as occuring in the beginning of a question (bqw), in the middle (mqw) of a question or at the end of a question (eqw).

A feature, Φ, is a binary function of the histories, h, and futures, f, of the problem being modelled. In answer type prediction, an example history stream is composed of the words of the question, QW, and the future is the class label, CL, associated with that question. A feature that uses the "who" word to decide that the label is PERSON is,

$$\Phi(h,f) = \begin{cases} 1 & \text{who} \in \text{QW \& PERSON} \in \text{CL} \\ 0 & \text{otherwise.} \end{cases}$$

In this application of maximum entropy, we propose such feature functions on instances of the training data where an error is made. The pool of such feature functions are ranked and the top features are selected in each iteration. The features of the maximum entropy model are ngrams of words (required to be adjacent) and bag of words where position is not important. Note that the organization of the data include bigrams which are upto distance 2, and additionally we have ngram features in the maxent model. The performance of the algorithm is shown in Fig. 2. Each feature type adds to the accuracy of the system and choosing 700 features with the"Mark Question Word" feature type achieves the lowest error rate of (9.05%) on a held out subset of the data.

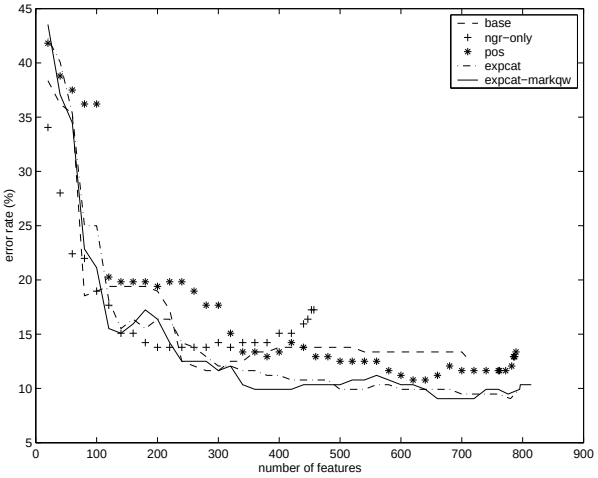

Figure 2: Answer Type Prediction Performance

Unigrams	What year did World War II start
Morphed{POS}	what{WP} year{NN} do{VBD} World{NP} War{NP} II{NP} start{NN}
Bigrams	what{wp} what{WP}_year{NN} what{WP}_do{VBD} ...
Expanded Hierarchy	what{WP} year time_period measure abstraction year{NN} do{VBD} ...
Mark Question Word	what_bqw year time_period measure abstraction year{NN} do{VBD} ...

Table 1: Data used for generating features for answer classification experiments

A peculiar feature of the architecture is that improvements in answer type prediction do not correlate directly with improvements in the overall score of the system. The reason is that parallel improvements must be made in the named entity marking as well as in answer selection in order to realize them in the overall system.

3 Information Retrieval

The purpose of the information retrieval module is to search the database of documents to select passages of text, containing information relevant to the query. The database used in TREC-9 has 978952 documents from several sources including AP Newswire, Wall Street Journal, San Jose Mercury News, Financial Times, Los Angeles Times, and the Federal Broadcast Information Service (FBIS). The database consists of approximately 2.8 GB of text, representing 524 million words.

Our information retrieval subsystem uses a two-pass approach. In the first pass, we searched an encyclopedia database. The highest scoring passages were then used to create expanded queries, applied in the second pass scoring of the TREC passages. The data preprocessing and relevance scoring techniques are similar to the ones applied in the TREC Ad-Hoc, SDR and CLIR participations (Franz and Roukos, 1998), (Franz et al., 1999).

Relevance scoring was based on morph unigram and bigram features, extracted from the text data in the following way: after the initial filtering, we tokenized the documents using a statistical tokenizer. The tokenized text was processed by a statistical part of speech(POS) tagger (Merialdo, 1990). Based on the spellings and the POS tags, the morphs were found by looking up the morph corresponding to a given word and POS tag in a table, e.g., the word "running" tagged as verb was converted into "run", whereas the same word marked as adjective was left unchanged. The words not found in the morph table were kept in their original form. All the words were case-folded after the morphological analysis was done. Hyphenated words were then split into their components.

We used a modified Okapi (Robertson et al., 1995) formula in the first-pass scoring. Unigrams and bigrams in the intersection of the query and document contributed a score of:

$$s = \frac{tf}{c_1 + c_2 \times \frac{dl}{avdl} + tf} \times \text{idf}, \qquad (1)$$

where tf is the term count for a document, dl is the document length, $avdl$ is the average length of the documents in the collection and idf is the inverse document frequency, computed as:

$$\text{idf} = \log(\frac{\text{N} - \text{n} + 0.5}{\text{n} + 0.5}),$$

where N is the total number of documents in the corpus and n is the number of documents containing a given n-gram. In Eq.(1) we used $c_1 = 0.5, c_2 = 1.5$ for unigram scoring and $c_1 = 0.05, c_2 = 0.05$ for the bigrams. The first pass score was a linear combination of unigram and bigram scores given by Eq.(1), with the unigram scores weight set to 0.8 and bigram scores weight equal to 0.2.

We computed first-pass relevance scores for 82,277 overlapping passages, each containing approximately 100 non-stop words, extracted from 18,910 encyclopedia articles.

Based on the first pass passage ranking, we constructed expanded queries using the local context analysis (LCA) technique (Xu and Croft, 1996). In the second pass scoring, the expanded queries are used to score 2,632,807 passages based on the TREC-9 Q&A corpus. The passages were selected to contain approximately 200 non-stop words.

	MRR
pass1, TREC	0.4605
pass2, TREC	0.4824
pass2, encyclopedia	0.5031

Table 2: Retrieval results.

Table 2 summarizes the information retrieval results on the 146 development test set questions described below. The performance is measured by the Mean Reciprocal Rank (MRR) (Voorhees and Tice, 1999) of the highest ranking passage containing the answer string among the top five passages. The first line of the table shows the result of first pass scoring using the TREC-9 Q&A database. The second line contains the result obtained with queries expanded using the TREC database. The last line of the table shows the result corresponding to the system applied in our official submission, with queries expanded using the encyclopedia database.

4 Named Entity Annotation

Named entity (NE) annotation is a markup of the text with the class information. As mentioned above, our classes correspond to the MUC classes due to the availability of training data for these classes. We used the text corpora available from the LDC to train the maximum entropy model.

Windows of +/- 2 words, morphs, part-of-speech (POS) tags and flags raised by pattern grammars for DATE, MONEY, CARDINAL, MEASURE, PERCENT, TIME, DURATION classes and dictionary hits, along with the two previous tags are created for each word. The window for predicting tag(0) is shown in Table 3. The window is the useful information given to the maximum entropy feature generation system to make features about the tag of the current word. The (-,+) signs give a clue to the feature functions about the relative position of this data to the word being tagged. Additionally, the (-2,-1,+1,+2) give position information to the feature functions. Each stream has a fixed vocabulary and n-grams from this vocabulary are created to be the features of the maximum entropy model. The training data is arranged to indicate a special category for beginning each named entity, for example Begin-PERSON, to find the boundaries of the named entity.

Words	w(-) w(-) | w(0) | w(+) w(+)
Morphs	m(-) m(-) | m(0) | m(+) m(+)
POS	p(-2) p(-1) | p(0) | p(1) p(2)
Flags	f(-2) f(-1) | f(0) | f(1) f(2)
Tags	t(-2).t(-1) t(-1) |

Table 3: Features used in the named entity model for predicting tag(0).

The system explores multiple NE hypotheses in parallel and keeps only those with high probability and proceeds with a beam-search algorithm to find the most likely path for the whole sentence. The performance of the named entity detector is comparable to the performance cited in (Borthwick et al., 1998) when training the maximum entropy algorithm on only annotated data. We omit the results here in the consideration of space, but note that in the analysis of the question answering system below, only 4 out of 64 errors are attributed directly to the named entity marking for the 250 byte system.

5 Answer Selection

We receive in this module the question, the class of the answer that the question seeks and a ranked set of passages (70) annotated with the MUC classes. The optimal sentence that answers the question is now sought. The TREC length constraints of 250 byte and 50 byte are then applied on the sentence.

The algorithm used in this module is listed here:

1. Each retrieved passage is split into sentences.
2. A window is formed around each sentence (window size is 3 sentences)
3. The following distances are computed: Matching Words, Thesaurus Match, Mis-Match Words, Dispersion, and Cluster Words. These are defined below.
4. The location or absence of the desired entities is noted in the score.
5. Each of these distances are weighted, the sentences ranked and the top 5 sentences

are then output.

The definition of the various distances are

Matching Words The TFIDF sum of the number of question words that matched answer words identically in the morphed space. (+)

Thesaurus Match The TFIDF sum of the number of question words that matched answer words using a thesaurus match using WordNet synonyms (Miller, 1990). (+)

Mis-Match Words The TFIDF sum of the number of question content words that did not match in this answer. (-)

Dispersion The number of answer words in the candidate sentence that occur between matching question words. (-)

Cluster Words The number of answer words in the candidate sentence that occurred adjacently in both the question and answer candidate. (+)

Each distance has a weight applied and the corresponding sign shown above attached to it. The score for an answer is the sum of the distances and the top 5 sentences are then output.

To select the 250 or 50 byte answer chunk from these sentences, the system identified the longest mismatched pieces between the answer string and the question. It then analyzed the answer and the question to find where the center of the match was using a subject-verb-object assumption of the sentence. The system then output either the subject or object portion whichever had the least matches with the question.

Answer selection as done above used ad-hoc and heuristic distance metrics to seek an answer. Future work by the authors will show how to treat these distance metrics as features and to develop a statistical model for answer selection for an open domain.

6 Development Set Analysis

We wanted to maintain the TREC-9 question database as a test set, but in order to do some post-evaluation analysis, we chose a subset of the questions as a development set for next year. There were two classes of questions in the TREC9 test: questions that had only one phrasing and questions that had more than one phrasing (rephrased). For example, the following questions form a set:

- Original Form:
- Rephrased:
- Rephrased:

We wanted 20% of questions of each class in the development test. The exact list of questions we used for our TREC-9 development test set are shown in Table 4. The variant questions we chose are shown in italics, and we added every seventh question skipping the ones in the above class to yield the 146 questions. A set of regular expressions (answer patterns) which detect the presense of the answer in a string was developed for the set using the judgements file provided by NIST.

201	203	209	210	217	220	224	231	238	242
245	252	253	259	264	266	273	275	280	286
287	294	297	301	308	315	319	322	329	330
336	341	343	350	352	357	363	364	371	374
378	385	392	393	399	411	412	413	420	434
453	454	456	458	462	469	473	476	483	484
490	495	497	504	506	511	517	518	525	528
532	539	546	550	553	560	561	567	572	574
581	583	588	594	595	602	605	609	616	623
627	630	637	638	644	649	651	658	660	665
671	672	679	682	686	693	700	*711*	*712*	*713*
714	*715*	*716*	*717*	*718*	*719*	*720*	*721*	*722*	*723*
724	*725*	*726*	*727*	*728*	*729*	*730*	*731*	*732*	*733*
734	*805*	*806*	*807*	*828*	*829*	*830*	*831*	*832*	*833*
834	*839*	*840*	*841*	*842*	*843*				

Table 4: Question numbers chosen for the TREC-9 development set.

The MRR for the entire system for the 250 byte system and the 50 byte system is shown in Table 5. The results of our system in the official evaluation and the development set evaluated using pattern rules are close in both the 250 and 50 byte numbers. Furthermore, the results indicate a 2% absolute MRR improvement using the encyclopedia source to expand the original questions.

Analysis of the components are shown in Table 6. An error is attributed to a component if

System	TREC9 results	DEV ENCL expansion	DEV TREC expansion
250 byte	0.457	0.437	0.417
50 byte	0.290	0.287	0.266

Table 5: MRR for TREC-9 and the chosen development set.

Component	Number of Errors	
	250 byte	50 byte
Answer Type	5 (3.4%)	7 (4.8%)
IR	19 (13%)	19 (13%)
NE	4 (2.7%)	5 (3.4%)
Answer Selection	36 (24.7%)	52 (35.6%)
System	64(43.8%)	83(56.8%)

Table 6: Component error rate for the TREC9 dev set for 250 byte system.

it is the first component that caused the failure working left to right in our system architecture. This analysis was carried out on the top-5 answer strings. Thus, a failure occurs if there is no answer produced by the system at all. Fixing this error though need not correct the final answer as it may invoke an error in a subsequent component. Answer selection is still seen to be the major cause of problems in our question answering system.

Another viewpoint is to see the effect of the system on the IR ranking results. This is shown below in Figure 3. Finding the 250 bytes from a passage that is of typical length 2.4K bytes shows some degradation, but further finding the 50 byte answer has considerable degradation. In Tables 7 and 8 we show the transition matrix for the rank from IR passages to the Q&A system results . Note that there are significant transitions between the IR rank and the Q&A rank, but that inspection of the final result in Figure 3 shows that overall system performance is similar to the performance of IR for the 250 byte system and degraded at 50 bytes. In Figure 3, we plot the number of queries which had an answer at rank 1..5 and indicate no answer produced by >5. These results we believe points to the possibility of making more improvements in answer selection by reranking the results.

Q&A rank	IR rank						Total
	1	2	3	4	5	5+	
1	29	9	5	3	2	5	53
2	10	2	1	0	0	0	13
3	2	2	1	0	1	0	6
4	1	1	0	1	1	2	6
5	2	1	0	0	1	0	4
5+	13	7	2	1	1	40	64
Total	57	22	9	5	6	47	146

Table 7: Rank transition matrix, IR ws Q&A, 250 bytes.

Q&A rank	IR rank						Total
	1	2	3	4	5	5+	
1	20	5	2	1	0	3	31
2	5	2	1	0	0	1	9
3	6	2	1	1	1	0	11
4	3	1	0	0	1	1	6
5	2	1	1	0	1	1	6
5+	21	11	4	3	3	41	83
Total	57	22	9	5	6	47	146

Table 8: Rank transition matrix, IR ws Q&A, 50 bytes.

7 Related Work

In the last two years, several efforts at question answering for open domain (Moldovan and et. al., 1999; Voorhees and Tice, 1999) and FAQ domains (Burke and et. al., 1997) have appeared. Our approach at question answering has been to follow the lead of the other participants in the TREC evaluation but base our components on maximum entropy modelling. We believe that corpus based systems allow technologies to be compared in a systematic approach, thus furthering the field of question answering.

8 Conclusion

We presented above our architecture and a component-wise evaluation of the system in the question answering problem. We developed maximum entropy formulations for both question/answer classification and named entity marking. The results presented above indicate a 2% absolute MRR improvement using the encyclopedia source to expand the original questions. The transition matrix of the IR to Q&A

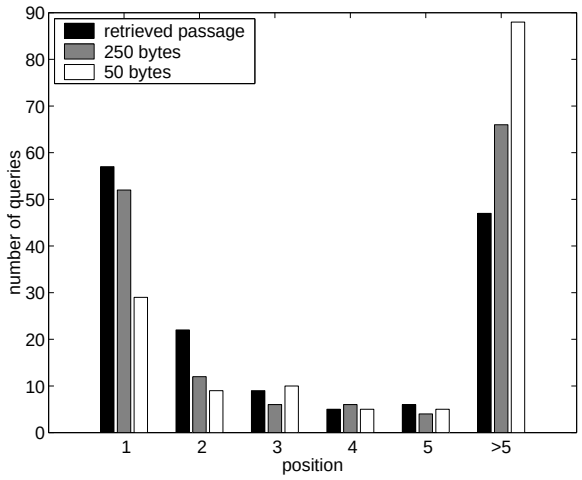

Figure 3: Development set performance comparing IR and Question-Answer ranking.

rank shows the effect of optimizing the various distance metrics used in answer selection. Future extensions of this work will utilize maximum entropy features in the answer selection process which will render the system completely trainable from examples.

9 Acknowledgement

This work is supported by DARPA under SPAWAR contract number N66001-99-2-8916. The authors thank Salim Roukos, Kishore Papineni and Todd Ward for making this work possible and helping us with their enormous expertise.

References

Academic Hallmarks. 1999. Knowledge Master. *http://www.greatauk.com.*

Adam Berger and Harry Printz. 1998. A comparison of criteria for maximum entropy/minimum divergence feature selection. *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, pages 97–106, June.

Adam L. Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1).

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. *Proceedings of the COLING-ACL 98, Sixth Workshop on Very Large Corpora.*

Robin Burke and et. al. 1997. Question answering from frequently-asked question files: Experiences with the faq finder system. *University of Chicago Technical Report TR-97-05.*

Nancy Chinchor. 1997. MUC-7 named entity task definition. *Proceedings of MUC-7.*

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1995. Inducing features of random fields. *Technical Report CMU-CS-95-144*, May.

M. Franz and S. Roukos. 1998. TREC-6 Ad-hoc retrieval. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication 500-240.

M. Franz, J. S. McCarley, and S. Roukos. 1999. Ad-hoc and multilingual information retrieval at IBM. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication 500-242.

B.F. Green, A.K. Wolf, C. Chomsky, and L.K. Baseball. 1963. An automatic question answerer. *Computers and Thought*, pages 207–216.

B. Merialdo. 1990. Tagging text with a probabilistic model. In *Proceedings of the IBM Natural Language ITL*, pages 161–172.

G. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

Dan Moldovan and et. al. 1999. Lasso: A tool for surfing the answer net. *TREC-8 Proceedings*, pages 65–73.

S.E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In D.K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*. NIST Special Publication 500-225.

Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. *TREC-8 Proceedings*, pages 41–63.

Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*