IBM's Statistical Question Answering System – TREC-10

Abraham Ittycheriah, Martin Franz, Salim Roukos
P.O.Box 218,
Yorktown Heights, NY 10598
{abei,franzm,roukos}@watson.ibm.com

Abstract

We describe herein the IBM Statistical Question Answering system for TREC-10 in detail. Based on experiences in TREC-9, we have adapted our system to deal with definition type questions and furthermore completed the trainability aspect of our question-answering system. The experiments performed in this evaluation confirmed our hypothesis that post-processing the IR engine results can achieve the same performance as incorporating query expansion terms into the retrieval engine.

1 Introduction

The TREC evaluations in question answering prompted many sites to develop technology to deal with open-domain question answering from real user queries. Our system focus is to create technology that can learn from question answer pairs sufficient rules and weights such that these can be used to find the answers to new questions. In TREC-9, we used a statistical algorithm for both answer tagging (predicting the class of the answer desired by a question), as well as named entity tagging (predicting the class of segments of text). Matching these predictions, as well as maximizing the overlap of question words to answer words yielded an answer to the question in our TREC-9 system. For TREC-10, we developed the following additional components:

- New and refined answer tag categories.
- Query expansion lists incorporated in answer selection.
- Focus expansion using *WordNet* (Miller, 1990).
- Dependency relationships using syntatic parsing.
- A maximum entropy formulation for answer selection (Ittycheriah, 2001).

These are described in the following sections and then we give some preliminary analysis of TREC-10 questions that were solved using these techniques as well as some crucial failures of our system. In the discussion below, we abuse the term TREC-9 meaning not the entire TREC-9 test but the first 500 questions which is the set of questions from the real TREC-9 test without the NIST reformulated questions.

2 Refining the Answer Tag Model

In our previous system, one of the predicted class is the unknown class, which we label as PHRASE. A comparison of the answer tags in three datasets is shown below in Fig. 1. Given the large number

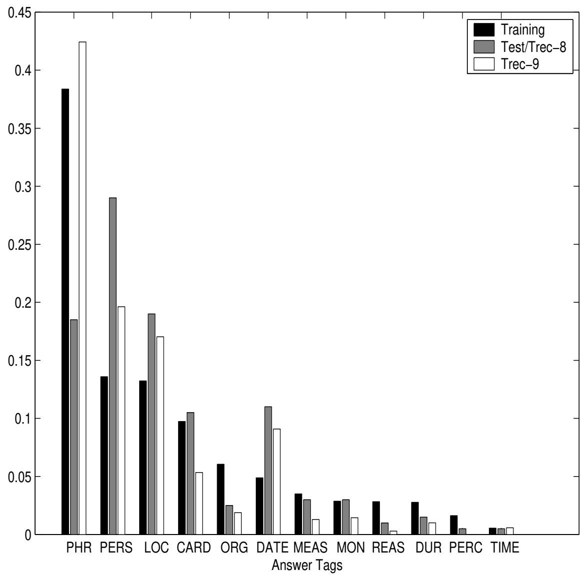


Figure 1: Histogram of Answer Tag Classes.

of questions being labelled as PHRASE, we set out for TREC-10 to increase the number of answer tags that we generated in the hopes of reducing the number of questions that were labelled PHRASE. Since both questions have to be annotated with the new answer tags as well as named entity material tagged similarly, it is a very expensive proposition to change the tag sets. We chose the following 31 tags as being a reasonable expansion of the categories used in TREC-9. The tags are broken along five major categories:

Name Expressions Person, Salutation, Organization, Location, Country, Product

Time Expressions Date, Date-Reference, Time

Number Expressions Percent, Money, Cardinal, Ordinal, Age, Measure, Duration

Earth Entities Geological Objects, Areas, Weather, Plant, Animal, Substance, Attraction

Human Entities Events, Organ, Disease, Occupation, Title-of-work, Law, People, Company-roles

Despite the increased number of answer tag classes, the percentage of PHRASE labelled questions has increased in TREC-10 to be 56% (280 out of 500 questions). These questions are dominated by "what is" or "what are" (232 out of 500 questions) which are mostly definitional type questions. Thus, the TREC-10 test is very similar to the TREC-9 test in terms of answer tags and definitional questions.

3 Incorporating Query Expansion in Answer Selection

Our information retrieval subsystem uses the same two-pass approach as our TREC-9 system. In the first pass, we search an encyclopedia database. The highest scoring passages were then used to create expanded queries, applied in the second pass scoring of the TREC documents. The data pre-processing and relevance scoring techniques are similar to the ones applied in the previous TREC evaluations (Franz and Roukos, 1998), (Franz et al., 1999). The expanded queries are constructed using the local context analysis (LCA) technique (Xu and Croft, 1996).

Examining the words used in the query expansion for IR, it was noted that often the answer words occurred on that list. Quantifying this observation, we measured the number of words intersecting the answer patterns for the TREC-9 test and found that 185 questions out of 500 questions had at least one word of its answer as part of the query expansion list. However, in our normal setting for query expansion we expand each query by a set number of words. Typical queries in this domain are five words and we add twenty words through the LCA expansion. For TREC-9, out of 10K words on the query expansion list only 297 words are actually part of the answer strings. In the sentence scoring portion of our system, we add to the sentence score half of the IDF weight of the expanding word. The top sentences are reranked by a maximum entropy based answer selection model and within this model we incorporate knowledge about the query expansion as a binary feature indicating the presence or absence of such expanding words.

4 Focus Matching

Question focus was defined in (Moldovan and et. al., 1999), and here we modify and state it as the word or sequence of words which optionally occur following a question word, which serves to indicate the answer tag. This notion has been used in our answer tag modelling previously (Ittycheriah et al., 2001), but in addition we used this notion in TREC-10 to get a refined sense for the broad categories as well as provide a substitute for answer tags in the case of PHRASE type questions. If the question has a focus, then answers who have hypernym (parent) or hyponym (child) relationship in *WordNet* are boosted in score. For example, a question such as *What river in the US is known as Big Muddy?*, the focus is derived to be *river* and in *WordNet* we see that the Mississippi River has a child relationship with river. The distance of the focus word to the nearest content question word is measured as is done for the named entities also. The focus score for the sentence i is then computed as

$$\mathrm{S}_{f,i} = (F + (3 - d_f) * D_f)$$

where F is the focus boost, d_f is the distance, and D_f the distance penalty for focus. At the best operating point, the focus boost is set to 10% of the total IDF weight of the question words and the distance penalty, D_f is set to 4.0. This score is added to the sentence score, which is described in (Ittycheriah et al., 2001).

5 Dependency Relationship Matching

Use of syntactic parsing has been used in information retrieval systems before, for example (Strzalkowski et al., 1997) shows an effective improvement in the precision rate using head-modifier pairs. Also, dependency structures on a syntactic parse is used in (Harabagiu and et. al., 2000) for deriving the logical form. The use of the dependency structure here will be to,

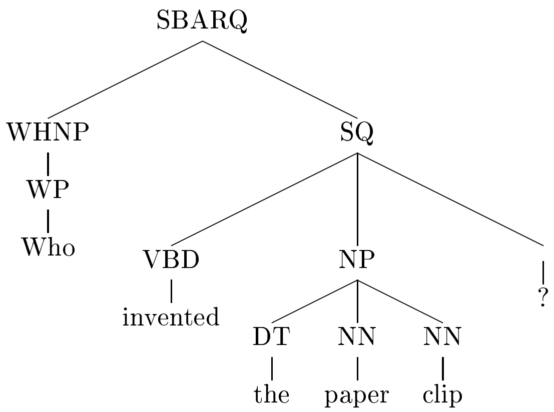
- constrain the match of branches
- to analyze what other extra words are in a dependent structure with the question words

- give credit to the proper named entity in a sentence

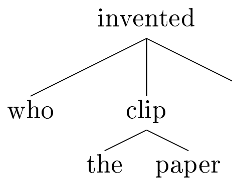
Using a parse structure gives the system the capability to score higher those answer candidates that have the words in a similar structure. Note though that the Cluster Words, Dispersion and Named Entity distance are performing similar functions though they are not explicit as using the parse structure. An example will motivate the use of the parse information.

5.1 Dependency Example

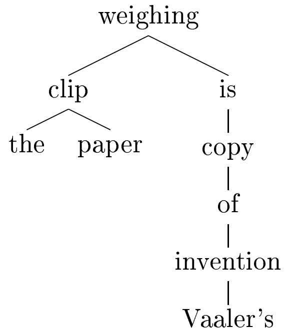
An example of a question in the TREC-9 corpus is, *Who invented the paper clip?* The parse for this question derived by the statistical parser is,



The dependency graph is,



The structure then reveals the requirements on the answer: specifically that the user is asking about a "paper clip" as opposed to "clips of paper". A strict *bag-of-words* technique assigns equal weight to both phrases. Given this dependency graph, it would be trivial now if the answer was essentially of the same form, but in this case the answer lies in the sentence *The paper clip, weighing a desk-crushing 1,320 pounds, is a faithful copy of Norwegian Johan Vaaler's 1899 invention, said ...* . A portion of the dependency tree derived for this sentence is,



While the entire parse tree can not be aligned per se, we present a method of doing partial matches below. Also note, that in the ideal answer, "Johan Vaaler invented the paper clip", the named entity that is desired will have a dependency to a word in the question. This is not in general true but for questions dealing with the defined set of entities it holds.

Additionally, consider the answer *"Like the guy who invented the safety pin, or the guy who invented the paper clip", David says.* In this answer to the above question, the segment "invented the paper clip" gets full credit. However, "David" gets no credit, since the dependency structure shows no relationship to the invention of the paper clip. Without a dependency structure, it is difficult from the surface text to determine which invented should get credit and whether a named entity match should count or not.

6 Trainable Answer Selection

In question-answering, a classification viewpoint to the problem would be to find answers, a, that maximize the conditional probability $p(a|q)$. Formulation of this model would allow us to search the space of answers and find the answer to the given question. However, this model is not tractable currently as the answer space is large and training data insuffient to directly model this distribution. Instead, we model the distribution $p(c|a,q)$, which attempts to measure the c, 'correctness', of the answer and question. We then introduce a hidden variable representing the class of the answer, e, (answer tag/named entity) as follows,

$$\begin{aligned} p(c|q,a) &= \sum_e p(c,e|q,a) \\ &= \sum_e \frac{p(c,e,q,a)}{p(q,a)} \\ &= \sum_e \frac{p(c|e,q,a)p(e,q,a)}{p(q,a)} \\ &= \sum_e \frac{p(c|e,q,a)p(e|q,a)p(q,a)}{p(q,a)} \\ &= \sum_e p(c|e,q,a)p(e|q,a) \end{aligned} \tag{1}$$

The terms, $p(e|q,a)$ and $p(c|e,q,a)$ are the familiar answer tag problem and the answer selection problem. Instead of summing over all entities, as a first approximation we only consider the top entity predicted by the answer tag model.

The distribution $p(c|e, q, a)$ is modelled utilizing the maximum entropy framework described in (Berger et al., 1996). The thirty-one features used in the model are described fully in (Ittycheriah, 2001) but the broad categories of features are presented here also. One feature represents the document rank as returned by our IR engine, twelve features are used on the sentence corresponding to the answer and the remaining eighteen are features on answer candidate strings.

The training data for the answer selection model is drawn from questions 251-450 of the TREC-9 test and the 200 questions of the TREC-8 test. The last 50 questions of the TREC-9 questions are used for a validation test of the model and the first 250 questions are used as a real test. This setup of the data is primarily motivated so that all data used in both training and test are generally available for all research sites.

6.1 Sentence Features

The sentence features allow the model to validate the original sentence ranking via a simpler weighted sum of scores approach. These features include for example the matching word score, thesaurus (*WordNet*) match, dependency arc match score, LCA expanding word score, and the cluster match score.

6.2 Entity Features

These features on the answer candidates reflect the finding of the desired entity and also help to overcome failures in named entity marking. The features incorporate knowledge about finding the entity, the focus being present, and whether the answer candidate has a proper noun, digit or date. The Candidate DB Focus is a feature that fires when a word that occurs next to the question focus is found in the answer candidate. In the example above, the feature fires for Mississippi because elsewhere in the text the word occurs next to "River". The feature is most useful when an answer satisfies the focus somewhere in the text and then subsequently the answer is used without the focus.

6.3 Definition Features

Definitional questions request an elaboration on a concept or given a concept requires the term of the definition. These questions are largely outside the scope of named entity analysis and focus methods. The questions are simple to answer when there are only few instances of the term and the term is used primarily in the definitional context. However, this is often not the case and since the questions typically have only one major phrase that is being defined, there is very little the match type features can do. Using a dictionary resource such as WordNet can aid greatly in answering these type of questions. The collection of these features isolates various types of matches from *WordNet* glosses to items found during LCA expansion.

6.4 Linguistic Features

Is-Relationship This feature fires when the answer candidate is either the subject or object of a form of the verb "be" and has all the question words following.

Apposition This feature applies to answer candidates which are immediately preceded or followed by a comma and then a group of matching question words are found. This is similar in function to the *Comma_3_words_score* of the LASSO system (Moldovan and et. al., 1999) although in this case its even more restrictive in requiring all question words to be present.

Subject-Verb/Verb-Object When the question has non-stop verb (meaning important and uncommon), and the answer candidate is either in subject or object position, this feature fires.

These thirty-one features were examined for the answer selection problem and there is no feature that completely separates correct answers from incorrect for all questions. The features are mostly real valued (for example the matching IDF sum of question words) and has to be quantized into some bins. The bin widths are uniformly spaced between the maximum and minimum value for a feature. The IDF features are quantized into four bins. The non-IDF features, such as "Digit Score" are already quantized since they only test for the presence of a numeric quantity in the answer chunk. As an example of the quantization process, the following matrix shows the distribution of the feature Matching_Word_Score among correct (COR) and incorrect (INC) chunks. Note that the label match_0 only indicates that the answer was in the lowest bin, not that the number of words that matched was zero.

	match_0	match_1	match_2	match_3
COR	425	46	55	384
INC	1407	267	282	1035

The selected features come from considering these features individually and up to order 4 combinations. The features sorted by their weight and then choosing the top 10 and bottom 10 are are shown in Table 1. The MRR versus the number of features is shown in Table 2.

These results indicate that the best performance is at about 168 features and it represents a 12.9% improvement over the baseline ranking. A few of the features are worth examination. First, the future "1" indicates a correct answer. All features with weight greater than 1.0 are associated with the "correct" class (1). This is because the prior distribution is strongly weighted for incorrect answers and the model needs a lot of features to overcome the prior distribution. The first feature indicates "CARDINAL_CARD", which means that the question desires an answer with a number and that the answer candidate has a numeric word. The feature "miss_0_candent_1_PERSON_PERSON_candarmatne_1" is a complex feature that requires that the answer candidate not have any missing words, it must have the desired entity, the desired answer tag and entity found must be PERSON and that the entity has a parse link structure to a question word. Intuition says that this probably indicates a correct answer and indeed it weights it with a weight greater than 1 and thus boosts the score if the model is predicting a correct answer. At the other end of the weights, features like "LOCATION_PERSON" indicate that if the desired answer tag is a LOCATION and the answer candidate has a PERSON, then it probably is not a correct answer (it weights the score down if this feature fires). The ability to interpret these features is a strength of the maximum entropy approach. The weight in general can not be evaluated, except that large deviations from 1.0 indicate either positive correlation with the future (when greater than 1.0) and negative correlation with the future (when significantly less than 1.0).

7 TREC-10 Results

In this years evaluation, we used our three submissions to evaluate the effect of various amounts of query expansion. The results are displayed in Table 3. Essentially, the results seem to indicate that increasing the amount of query expansion in our IR engine results in poorer overall performance. These experiments were done to determine whether a basic search engine should be modified to improve question-answering and the results indicate that at least for our search engine (Franz et al., 1999), question answering can be performed as a post-processing of the IR results. These results represent a 34.5% improvement over our 50 byte results in TREC-9.

8 Development Set Analysis

In this section, we focus the analysis on the system ibmsqa01a. There were surprisingly 30 out of 500 questions which only one system answered correctly. Of these, 7 were answered by the ibmsqa01a system. In order to reserve a true test set for next years evaluation, we chose to look only at the first 200 questions. Two examples from this development portion where our system produced answers and there was no other systems with correct answers are shown below.

Q: What do you call a newborn kangaroo?
A: 960 Q0 AP891022-0031 2 4.6789 ibmsqa01a - inch - tall baby _ called a joey _ followed .

Q: How fast is alcohol absorbed?
A: 1065 Q0 SJMN91-06037052 3 8.6748 ibmsqa01a one hour to metabolize one ounce of alcohol .

There were however 37 questions that our system blundered on, by which I mean that we produced no answer when 10 systems were able to get the correct answer. An example of such a question is "What is caffeine?", where our system got the word "coffee" from the definition in *WordNet* and produced answers which contained the word. The correct answer, 'alkaloid', is also in the *WordNet* definition, but the system preferred answers with coffee. Another example of a question where the system failed was "How many liters in a gallon?" to which our system produced as the best sentence "There are 3.8 liters in a gallon.", but then during the answer extraction (50 byte), this answer was thrown away because answer tag we searched for was CARDINAL and the answer contained a MEASURE. This can be considered as a failure in the answer tag selection, but it could be corrected by the answer selection if we had sufficient examples in our training data where such a mapping was desired.

9 Conclusions and Future Work

Our statistical question answering system showed significant improvement over the year (34.5%). This improvement was largely dominated by the inclusion of query expansion in answer selection and modest improvements were obtained by using a statistical algorithm for answer selection. The trainability of the answer selection still suffers from lack of training material so for our next system we are attempting to increase the training set by an order of magnitude more questions.

History	Future	Weight
CARDINAL_CARD	1	2.67586
GEOLOGICALOBJ	1	2.04765
candmaxmat_2	1	2.04757
MEASURE_MEASURE	1	1.80068
arcmat_0_candnumglossexp_1	1	1.61033
canddate_1_candarmatne_1	1	1.54951
miss_0_candent_1_PERSON_PERSON_candarmatne_1	1	1.53813
clusterscore_1_prevscore_1_no_ne_match_PHRASE_X	1	1.52065
candfoc_1_candnumdefnexp_1	1	1.42604
match_0_miss_0_candnumdefnexp_1	1	1.40497
miss_0_candfoc_1_candarmatne_1_candnumdefnexp_1	0	0.62682
arcmat_0_ne_map_match	1	0.626688
exact_ne_match_candmaxmat_1	0	0.570918
no_ne_match_canddist_2	1	0.570399
arcmat_0_clusterscore_1_no_ne_match_candarmatne_1	1	0.560376
TITLE_WORK_X	1	0.516318
ne_map_match_docrank-1	1	0.505354
DATE_X	1	0.401585
PERSON_X	1	0.369949
LOCATION_PERSON	1	0.302307

Table 1: Maximum Entropy features selected for answer selection.

.

Number of Features	Baseline	48	72	96	120	144	168	192
TREC9 MRR (q450-500)	0.458	0.487	0.487	0.489	0.496	0.509	0.517	0.496

Table 2: Maximum Entropy Performance versus number of features.

.

		Strict		Lenient	
System	Description	MRR	Num Missed	MRR	Num Missed
A	No query expansion in IR, Ency query expansion in Answer Selection	0.390	218	0.403	212
B	Ency query expansion in IR and Answer Selection	0.390	220	0.403	215
C	Ency and *WordNet* query expansion in IR and Answer Selection	0.375	231	0.388	224

Table 3: Performance on TREC-10.

.

10 Acknowledgement

This work is supported by DARPA under SPAWAR contract number N66001-99-2-8916.

References

Adam L. Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

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.

Sanda Harabagiu and et. al. 2000. Falcon: Boosting knowledge for answer engines. *TREC-9 Proceedings*, pages 50–59.

Abraham Ittycheriah, Martin Franz, Wei-Jing Zhu, Adwait Ratnaparki, and Richard Mammone. 2001. Question answering using maximum entropy components. *The Second Meeting of the North American Chapter of the Association of Computational Linguistics, Pittsburgh, PA*, pages 33–39.

Abraham Ittycheriah. 2001. *Trainable Question Answering Systems.* PhD Thesis, Department of Electrical and Computer Engineering, Rutgers - The State University of New Jersey.

G. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

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

Tomek Strzalkowski, Fang Lin, Jose Perez-Carballo, and Jin Wang. 1997. Building effective queries in natural language information retrieval. *ANLP*, pages 299–306.

Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *Research and Development in Information Retrieval*, pages 4–11.