Learning How to Answer Questions Using Trivia Games

Gideon S. Mann
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland 21218
gsm@cs.jhu.edu

Abstract

In this paper we examine a sentence comprehension task: given a question, and an extended sentence known to answer that question, the goal is to extract the short answer to the question. As an initial solution, a novel robust statistical model is presented which combines the semantics of the expected answer with the expected context within which the answer will be found. Two distinct trivia game databases, with no additional annotation, are used to train and test the model.

1 Introduction

Recent work in automated question answering has suggested that extracting precise short answers is more difficult than extracting full sentence or passage length ones.[1] Also, retrieving answers from smaller document spaces may be more difficult than retrieving answers from larger ones, if the smaller spaces have less redundant coverage of potential answers.[2] This paper examines the problem of extracting short answers from a single sentence, where there is virtually no redundancy. The focus on word-level analysis of a single sentence may uncover aspects of semantics which have heretofore been masked in sentence granularity answers.

To approach this sentence comprehension task, we introduce a novel robust statistical model. One of the barriers to building statistical models for question answering has been the lack of suitable training data, and the cost associated with annotating data when available (Ittycheriah et al., 2001). Here we explore the use of raw, unannotated "found" data from the Web and present the first system that exploits whole sentence explanations in training a short answer extraction model.[3] An example of these data is shown in Figure 1.

The model we propose integrates information from two different sources: semantic class preferences (restrictions on the type of answer expected, e.g. a person or a plant) and answer context (words likely to appear near the answer). Answer context is exploited using a new local alignment method. After training, the complete model achieves 70% match accuracy on one dataset, and 59% on another.

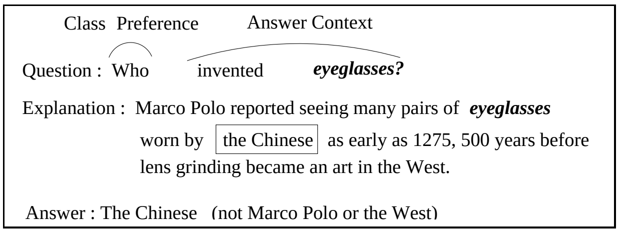

Class Preference
Answer Context
Question : Who invented *eyeglasses?*
Explanation : Marco Polo reported seeing many pairs of *eyeglasses* worn by the Chinese as early as 1275, 500 years before lens grinding became an art in the West.
Answer : The Chinese (not Marco Polo or the West)

Figure 1: TS Trivia Database Entry #42764. Both class preference and context must be used to extract the correct answer.

2 Trivia Games as Training Data

A practical problem in developing a statistical model for question answering is the lack of training data. We explore the use of unmodified, unannotated trivia games as data, as suggested in (Mann, 2001).

With the increasing mass appeal of the Web, trivia games are becoming big business. The general public submits questions, and trivia game companies award prizes to those who correctly answer the most questions, and in the process profit from advertising. Some trivia databases are quite large, containing up to two hundred thousand entries (Ford, 1999).

In this paper we use two trivia databases as main resources: "Phishy" – PH (MacDonald, 1998) and "TriviaSpot" – TS (Trivia Machine Inc., 1999). PH has approximately 5,000 questions, each with the correct answer. TS is larger (more than 50,000 questions), but only a small part (11,000 questions) was made available for this research. Each TS database entry, along with the correct answer, includes three wrong answers and in many cases an explanation. The explanations in TS vary in content. Some are justifications for the answer as in Figure 1. Others

[1] In the TREC-8 (1999) conference, performance on the 50-byte track was 10%-20% lower than performance on the 250-byte track for systems tested on both.

[2] Light et al. (2001) shows that the TREC corpus has more potential answers per question than the CBC reading comprehension task, and that the prescence of more potential answers leads to higher average system performance.

[3] Berger et al. (2000) uses explanations for modeling whole sentence retrieval.

provide additional information (e.g. "*Leonardo Da Vinci outlined the idea of contact lenses in 1508.*") or refute wrong answers (e.g. "*Franklin wore glasses, but didn't invent them.*"). Both of these databases contain questions and answers written by many different people, and there is no guarantee that the answers to the questions are correct.

Trivia games like these are an appealing source of data for those interested in question answering. First, they cover a wide domain of knowledge and are grammatically complex. Second, the questions typically ask for simple facts whose answer is often only a word or two, which is similar to other question answering tasks. Finally, trivia databases encode a tremendous amount of semi-structured information. In this paper we extract two types of knowledge from this heterogeneous database: class preference for semantic tags, and mixture parameters for our final model. These databases doubtlessly contain additional information.

While the TS database contains explanations, the PH database does not. To generate explanations for the PH questions, we searched the Web using the Google search engine with both the question words and the short answer as the query terms. We retrieved the top 100 ranked pages for each query and segmented the sentences automatically. From those sentences, we selected as explanations the sentences which had both the highest word overlap with the question and contained a word in the short answer.

As a result of this search procedure, we collected nearly 22,000 explanations for approximately 3,000 questions. Of course, the explanations were quite noisy. Some, though they included the short answer terms, did not have enough information to conclude that the short answer in fact answered the question. Many explanations were ungrammatical or odd mixtures of sentence fragments and spurious punctuation. Since the explanations were collected automatically, in some cases we found multiple explanations for the same question. For evaluation purposes, a sample of 1,000 sentences were randomly selected from each corpus.

3 Using and Detecting Class Preferences in the Question

Many questions give a clear class preference on the types of answers they expect. For example, the question "*The Star of Africa is what type of gem?*" strongly prefers the answer to be a gem. Intuitively, class preferences can be described as words in the question which are somewhat unlikely to appear in the answer sentence, and instead serve to restrict the types of entities suitable as answers. Sometimes, these preferences can be found in the WH phrase itself (e.g. "Who", "Where", "What type of fruit"). In other questions, more useful preferences can be found in different locations. For example in *"What is the profession of James Herriot"*, profession is the class preference.

Class preferences fall into two broad categories: preferences for proper nouns and for common nouns. Proper nouns exhibit great plasticity in use, sometimes even ranging over lexical elements which are primarily function words (e.g. the rock band "The The"). In contrast, common noun classes, while open to novel constructions, are much more static. One example of a common noun class preference is for color names, which are dominated by a static fixed set, with rare additions in popular usage (e.g. "teal").

Since these two classes of preferences exhibit such different characteristics, different types of processes may be used to deal with both. In order to detect proper nouns, a dynamic algorithm which tags tokens according to a set of contextual and intra-word features (Breck et al., 1999; Cucerzan and Yarowsky, 1999) must be used. For common nouns, a large, static hand-crafted knowledge base is more appropriate. The next two sections describe methods to handle both types of categories.

3.1 A Mutual Information (MI) Model for Proper Nouns

The main focus of work in question answering has been on identifying answers which are proper nouns, where most methods predict likely semantic tags from WH heads or phrases (Prager et al., 1999). In these schemes a "Who" question would predict "People" and perhaps "Organizations". Typically these correspondences have been manually identified. Mann (2001) presents the following method for learning these correspondences using unannotated training data which results in a more flexible and effective match module.

Formally, given a question q with a unique WH head c, and an explanation E which is a set of words w, where each word w has a distribution over semantic tags st, the model picks

$$\begin{aligned}
\hat{w} &= \operatorname*{argmax}_{w\in E} I(q;w) \\
&\approx \operatorname*{argmax}_{w\in E} I(c;w) \\
&= \operatorname*{argmax}_{w\in E} \frac{P(w|c)}{P(w)} \\
&= \operatorname*{argmax}_{w\in E} \sum_{st} \frac{P(w|st)P(st|c)}{P(w)} \\
&= \operatorname*{argmax}_{w\in E} \sum_{st} \frac{P(st|w)P(st|c)}{P(st)} \\
&= \operatorname*{argmax}_{w\in E} \sum_{st} P(st|\mathrm{w})\ I(c;st) \qquad (1)
\end{aligned}$$

Class	Sem Tag	I(WHO ; st)
Who	Person	6.166
	Location	2.186
	Name	1.174
	Organization	1.333
	Country	0.583
	Date	0.402
	Time	0.311
	Title	0.210
	City	0.198
	Age	0.180
	Volume	0.130
	Noun Group	0.021
	Other	0.010
	Duration	0.011
	Quantity	0.039
	Length	0.007

Table 1: Tags having high Mutual Information with WHO, I(WHO ; st), as estimated from the Phishy (PH) trivia database

This derivation shows that we can approximate the Mutual Information between a question and a word, I(q;w), assuming the independence of the question class and answer given the semantic tag, using the probabilities in (1). To train the model, we need to estimate I(c;st) – how much a class prefers a semantic tag. Table 1 gives an example of the type of information learned by the MI model.

The above probabilities can be estimated using a trivia database that contains a large number of questions and answers. The method follows:

- For each question, identify the question class.
- Apply a semantic tagger[4] to the trivia database to generate $P(st|w)$. Thereby, a distribution over possible semantic tags for each word is estimated.
- Estimate:
$$P(st|c) = \sum_{w} P(st|w)P(w|c)$$

3.2 Ontological and Exact Match Models for Common Nouns

This section deals with common noun class preferences, where static knowledge bases can be used. Common noun class preferences are detected in two ways. First, the WH phrase is scanned. The question *"On what street is the Bank of England located?"* has a class restriction embedded in the WH phrase itself.[5] If no class preference has been found directly within the WH phrase, and the question uses a copular construction (e.g. "What is..."), the head of the first NP is used as a class preference. In the question, *"What is the color of sapphires?"*, color would be appropriately identified as a class preference.

These class preferences are used in two ways. First, noun phrases which include the word verbatim are chosen. Thus, "Threadneedle Street" would be detected as an answer to the first question in Section 3.2. Second, the sentence is scanned for any words which are children of the class in the Wordnet ontology (Miller, 1990). For example, "blue" would be detected as a type of color in response to the second question above.

3.3 Improving WH Phrase Segmentation

Segmenting the WH phrase and parsing the question correctly are key prerequisites for determining and using class preference information. Collins (1999) has demonstrated high performance on parsing plain English sentences and is an obvious choice for parsing questions. However, the Collins parser trained on the Penn Treebank does not provide high-quality WH phrase segmentation. Figure 2 shows a typical WH phrase segmentation error, where there is no node which subsumes only the WH phrase. Since the recovered WH phrases are incomplete, the WH phrases can be said to be fragmented. The misparses are caused by the paucity of questions in the Penn Treebank and the simplicity in form of the ones that do exist. In particular, there are no instances of a WH phrase parsed as (WHNP (WDT NP)).[6] This dearth of variety severely limits the ability of a parser trained only on the Treebank parser to find correct question parses.

In order to correct for this problem and to clean up the parses (to make their structure easier to interpret), 300 questions typed by users into an online information retrieval engine were collected[7] and hand-parsed for use as training data. These questions were randomly selected in equal proportions from these categories:

- 'What' : What kinds of jobs can you get with different degrees?
- 'Which' : Which country borders Belize?
- 'In what' : In what years did "My Fair Lady" and West Side Story" open on Broadway in NYC?
- 'In which' : In which countries do Habu snakes live?

[4] In our experiments we used Phrag (Breck et al., 1999), a Named Entity tagger.

[5] The WH head and "type of" and "kind of" phrases are excluded as potential common noun class preference indicators.

[6] An example of this phrase would be (WHNP (WDT which) (NP (JJ major) (NN river))).

[7] The questions were collected by John Henderson, MITRE, 2001.

(TOP (FRAG (IN In) (SBAR (WHNP (WDT which)) (S (NPB (NNP Sea)) (VP (VBZ is) (NPB (NNP Cuba) (. ?)))))))
(TOP (SINV (WHPP (IN In) (WHNP (WHNP (WDT which)) (NPB (NNP Sea)))) (SQ (VBZ is) (NPB (NNP Cuba) (. ?)))))

Figure 2: A mis-segmented question, and its corrected version

- Embedded 'what' : Beethoven's Sixth Symphony is also known as what?
- 'What do' : What do the initials CNN stand for?

After annotating these extra questions and retraining the Collins parser on the annotated questions and the Penn Treebank, the number of fragmented WH phrases dropped from 100 to 18 for the PH data set, and from 183 to 70 for the TS set.

3.4 Experiments

We tested the performance of the ontological and exact match approaches on the two datasets by themselves and in conjunction with the class/semantic tag model (Eq. 1). As a naive baseline, we present a system which chooses a word at random from within the sentence. The accuracy of the random system is evaluated as (#correct one word answers in the sentence)/(#total candidate answers). For all systems, we exclude words found in the question as well as stop words from candidate answers. In all of the experiments reported, a system is said to have answered a question correctly if the single word it retrieved is in the set of words the system accepts as the answer.

		PH	TS
ontological match	coverage	35%	13%
	accuracy	89%	71%
exact match	coverage	10%	2%
	accuracy	64%	35%

Table 2: Common Noun Class Preference Methods in Isolation

	PH	TS
random	30%	22%
I(c;st)	51%	52%
+ exact match	51%	52%
+ ontological match	**67%**	**56%**
+ both	65%	56%

Table 3: Accuracy of Systems which only use Class Preferences

Table 2 presents performance results for questions in which common noun class preferences, either from ontological matches or exact matches, could be found. The performance of both methods shows a trend wherein the PH set has higher coverage (more common noun class preferences can be found) and higher accuracy, while TS has both smaller coverage and lower accuracy. This might be in part explained by biases in the different test sets. Despite the differences, the ontological match accuracy is high (over 70%) on both test sets.

Table 3 shows the overall performance of the system, when the ontology and exact match components are used before the MI proper name model. The results demonstrate that adding the ontological model can greatly improve the performance of the whole system.

4 Using Answer Context with a Local Alignment Method

The methods described above have only used class-based approaches for finding potential answers in sentences. These methods ignore potential information from shared structure which might augment the performance. Prior methods for using structural information have relied on picking out information from parses (Hovy et al., 2001) or complex inference chains (Harabagiu et al., 2000) in order to extract the correct words from the sentence.

In this paper, we examine an 'informal semantics' approach, which is at the other end of the spectrum. Without building an intermediary semantic representation, we use matching structure directly to indicate which words are best matches.[8] The model we developed first finds **anchors**, words that appear in both the question and explanation. Once anchors are found, for any given tuple (anchor α, word w, and WH head c), three features are use to determine the quality of the local alignment:

- $\mathbf{de}(\mathbf{w}, \alpha)$, the distance in the explanation from w to α: Larger distance between the answer and

[8] One idea on how to use contextual information is to treat it as an alignment problem, which could be modeled in similar ways to statistical machine translation alignment (Brown et al., 1993). Initially, we looked at a whole sentence alignment model, using a Stochastic Inversion Transduction Grammar, SITG, (Wu, 1997) which aligned an entire question with an entire explanation. This approach had poor performance for several reasons. The lexical change from sentence to question is often quite elaborate, and taking it into account in a general way is quite difficult. On top of this, the SITG model has limitations in terms of the kinds of transpositions it allows. In particular, phrases like "X is Y" require two inversions to turn into "Y is X". What is needed is a grammar which handles these kinds of common phrasing alternations more effectively.

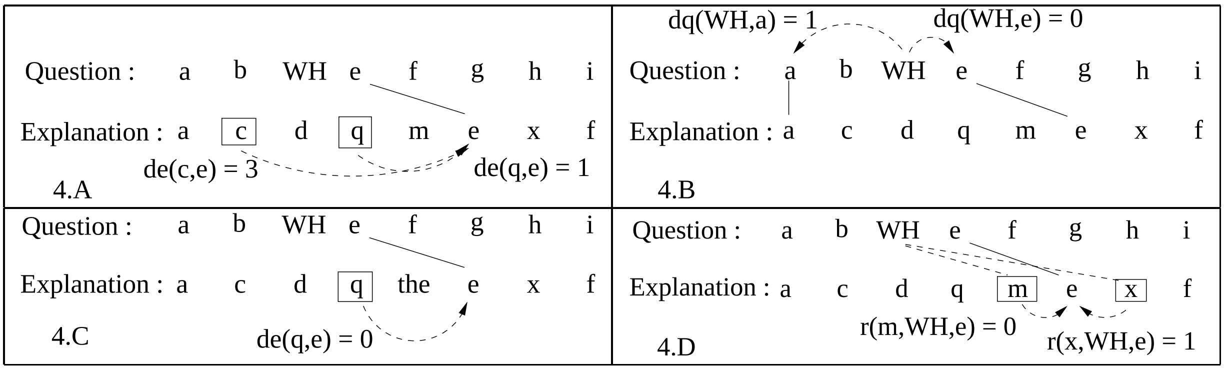

Question : a b WH e f g h i
Explanation : a c d q m e x f
de(c,e) = 3
de(q,e) = 1
4.A
dq(WH,a) = 1
dq(WH,e) = 0
Question : a b WH e f g h i
Explanation : a c d q m e x f
4.B
Question : a b WH e f g h i
Explanation : a c d q the e x f
4.C
de(q,e) = 0
Question : a b WH e f g h i
Explanation : a c d q m e x f
r(m,WH,e) = 0
r(x,WH,e) = 1
4.D

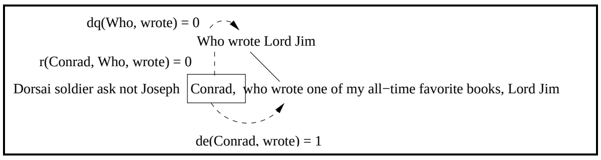
Figure 3: Features of the Local Alignment Model: (4.A) $de(w, \alpha)$, distance from the $\boxed{\text{answer}}$ (boxed) to the anchor (4.B) $dq(c, \alpha)$, distance from the WH head to the anchor (4.C) $de(X, \alpha)$ ignores determiners (4.D) $r(w, c, \alpha)$, whether the WH head and the answer are on the same side of the anchor

the anchor indicates a weaker relationship. Distance is measured as the number of intervening content words.

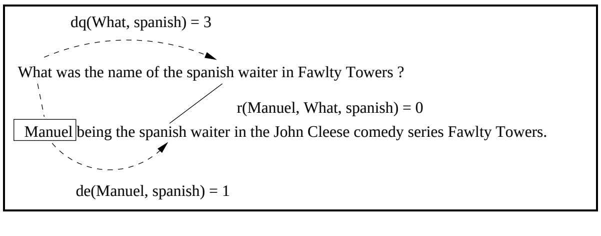
- $\mathbf{dq}(\mathbf{c}, \alpha)$, the distance in the question from c to α: The closer to the WH phrase an anchor is, the better it is for judging local alignment. Distance is measured as above.
- $\mathbf{r}(\mathbf{w}, \mathbf{c}, \alpha)$, whether w and c are on the same side of α : If they are on the same side, that suggests that the underlying semantic relationships in the two sentences are similar.

Figure 3 illustrates the features of this model.

In some cases, this local alignment method mimics what a more structured semantic representation would do, finding the head of the verb to which the answer is the object:


dq(Who, wrote) = 0
Who wrote Lord Jim
r(Conrad, Who, wrote) = 0
Dorsai soldier ask not Joseph Conrad, who wrote one of my all−time favorite books, Lord Jim
de(Conrad, wrote) = 1

In other instances, the aligned word corresponds to a relevant word in the question:


dq(What, spanish) = 3
What was the name of the spanish waiter in Fawlty Towers ?
r(Manuel, What, spanish) = 0
Manuel being the spanish waiter in the John Cleese comedy series Fawlty Towers.
de(Manuel, spanish) = 1

The alignment model is a robust method of extracting semantic relationships from a sentence. A more structured representation, using the parse of the sentence for example, might be able to provide the same information. However, a more structured representation takes a performance penalty commensurate with the performance of the parser, requires overhead to write rules to extract information from the parse, and necessitates a more complex mixture with a statistical class model. Nevertheless, future work might explore the use of higher-level structural information.

4.1 Creating a Unified Local Alignment Model

The prior sections have described a set of features $(I(c; w), de(w, \alpha), dq(c, \alpha), r(w, c, \alpha))$ which might help distinguish correct answers. In order for them to be used effectively, they have to be combined. In order to do this, we first model each feature as $P(a(\text{word})|\text{feature})$, where $a(w)$ is a function which indicates whether w is in the short answer to the question. For example:

$$P(a(w)|de(w, \alpha)) = \frac{c(a(w), de(w, \alpha))}{c(de(w, \alpha))}$$

where c is the number of times that feature appeared for each explanation/question pair, using any possible anchor. With these probabilities estimated, we smooth the graph, and build a model from which we can then interpolate new values. We also enforce a decreasing monotonicity condition, such that $P(a(w)|f_1) > P(a(w)|f_2)$ iff $f_1 < f_2$. We do this for the two distance measures (distance of the WH phrase from the anchor, and distance of the answer candidate from the anchor) and for the class/semantic tag match statistic as well. $P(a(w)|r(w, c, \alpha))$ can be calculated in a straightforward manner, and requires no smoothing (since it has only two binary values). The probability estimates derived from the two trivia databases are shown in Figure 4.

We use a mixture model to estimate the combined probability of a particular word being the answer. Formally, given an explanation e, and a question q with the WH head c, for any word w the model

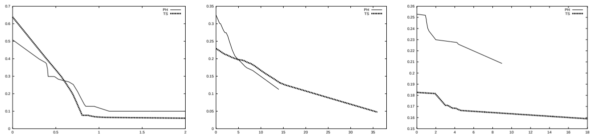


Figure 4: From left, estimation of: $P(a(w)|I(c;w))$, $P(a(w)|de(w,\alpha)$, $P(a(w)|dq(c,\alpha))$. TS in **bold**.

estimates its probability of being the answer as :

$$\begin{aligned}\hat{P}(a(w)|c,q,e) &= \max_{\alpha}\ P(a(w)|q,\alpha,e)\\ &= \max_{\alpha}\ [\lambda_1 P(a(w)|I(c;w))\\ &\qquad + \lambda_2 P(a(w)|de(w,\alpha))\\ &\qquad + \lambda_3 P(a(w)|dq(c,\alpha))\\ &\qquad + \lambda_4 P(a(w)|r(c,w,\alpha))]\end{aligned}$$

We use one trivia database to train the mixture parameters by exhaustively searching the space for the parameters which achieve the highest performance. We then take the optimal parameters and use them to test accuracy on the other trivia dataset.

4.2 Undoing WH Phrase Movement

Before attempting the local alignment, we preprocesses the questions to undo WH phrase movement. WH movement can be described through the following generative process: start with a sentence "*Marc likes Spot.*". Next the desired NP phrase answer is turned into a WH phrase ("*Marc likes what?*"). Finally, the missing WH phrase is moved to the beginning of the sentence ("*What does Marc like?*") and a tensed verb is added. This syntactic movement has been extensively studied in the linguistics and parsing communities. For example, Collins (1999) describes a method for discovering the trace of the "original" WH phrase position.

Clearly, WH movement is problematic for a system which uses local alignment, since the relative distances and locations within the questions do not match what they would be within the explanation. In order to compensate for these changes, we explore a simple method for detecting and resolving common movement in questions. In particular, we use the following characteristics to detect when a WH move might have occurred :

1. Trailing prepositions:

   > What were George Washington's false teeth made of?

2. Certain 'do' constructions:

   > What did the Taft-Hartley Act of 1947 attempt to regulate?

3. Initial prepositional phrases:

   > In what field of study would you find flying buttresses?

4. Simple Templates:

   > What was Gary Numan's first album on the Beggars Banquet label called?

5. Initial WH head adverbial phrases:

   > How is Frederick Austerlitz better known?

Once a question has been detected as likely to have involved movement, the WH phrase is removed and placed at the end of the sentence. The end is not always the trace position, but in our test sets it was for an overwhelming number of questions. With our retrained WH segmenter and question parser (Section 3.3), this method for unfolding WH-movement improves overall system performance.

4.3 Experiments

To test the local alignment model, we took the two sets of 1,000 sentences used for the class preference experiments and the class-preference models initially trained. We selected one of the databases as training set and the other as test set. We estimated $P(a(w)|f)$ for each of the features (the distance and MI features) on the training set, and optimized the mixture parameters on the training set. The optimal parameter values were:

	λ_1	λ_2	λ_3	λ_4
	I(c;w)	$de(w,\alpha)$	$dq(c,\alpha)$	$r(w,c,\alpha)$
PH	0.1	0.1	0.75	0.05
TS	0.15	0.55	0.1	0.2

With the optimized weights and trained $P(a(w)|f)$ models, we tested the models on the other database. The results shown in Table 4 demonstrate that the local alignment gives a consistent gain of 3-4% for each of the test sets.

Test Corpus	Class only	Class + Context
PH (c/st)	51%	55%
(c/st + ontology)	67%	**70%**
TS (c/st)	52%	56%
(c/st + ontology)	56%	**59%**

Table 4: Accuracy Improvement from Contextual (Local Alignment) Method

Figure 5 shows the precision/recall trade-off for the final model, using the final output mixture probability. The graphs illustrate that while there is a correlation between the estimated probability and the real probabilty of the answer being correct, there are still major deficiencies in the model. Perhaps part of the reason behind the disparity between the

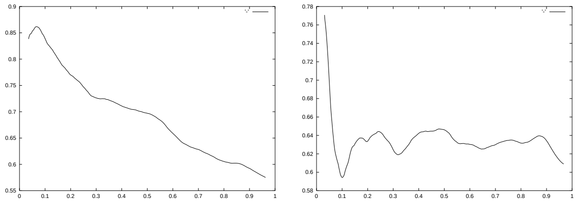


Figure 5: Precision/Recall (c/st onto) TS, PH

two graphs is that the explanations in TS were generated in response to the questions while the explanations for PH were extracted from the world wide web, and hence very noisy.

5 Conclusions

In this paper, we presented a novel robust statistical model for a sentence comprehension task. The local alignment method presents a new robust way to match answer contexts from the question and the explanation. Combining the local alignment with a statistical model of class/semantic tag match model and an ontology match module, we were able to reach high levels of performance on this task (70% on the Phishy data and 59% on the TriviaSpot data).

Through the training of the mixture model for combining local alignment features and the class/semantic tag match, we demonstrated the value of explanations for statistical estimation of intra-sentential features. The use of explanations for training this model suggests that (question, explanation, answer) tuples may be useful in training future question answering systems.

In investigating this sentence comprehension problem, we took an informal semantics approach. This operational semantics may allow the system to see aspects of meaning which are lost in truth-functional semantics. In general, this sentence comprehension task presents an opportunity for empirically testing semantic representations, without the overhead needed to complete the full information retrieval task required in question answering (i.e. no document or sentence retrieval needed). Deficiencies in the sentence comprehension module might be masked by a very effective document retrieval system, but exposed under this more constrained test.

References

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

E. Breck, J. Burger, L. Ferro, D. House, M. Light, and I. Manni. 1999. A Sys Called Qanda. *Proceedings of the Eighth Text Retrieval Conference.*

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R. Mercer. 1993. The Mathematics of Statistical Machine Translation. *Computational Linguistics*, 19(2):263-311.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. thesis, University of Pennsylvania.

S. Cucerzan and D. E. Yarowsky. 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, pages 90–99.

G. MacDonald (ed). 1998. Phishy web trivia. *www.phishy.net/trivia.*

T. Ford (ed). 1999. Fun trivia.com : the trivia portal. *www.funtrivia.com.*

S. M. Harabagiu, M. A. Pasca, and S. J. Maiorano. 2000. Experiments with Open-Domain textual Question Answering. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING 2000)*, pages 292–298.

E. Hovy, U. Hermjakob, and C-Y. Lin. 2001. The Use of External Knowledge in Factoid QA. In *Proceedings of the Tenth Text Retrieval Conference.*

A. Ittycheriah, M. Franz, W-J. Zhu, and A. Ratnaparkhi. 2001. Question Answering Using Maximum Entropy Components. *Proceedings of the Second Meeting of The North American Chapter of the Association of Computational Linguistics*, pages 33–39.

M. Light, G. S. Mann, E. Riloff, and E. Breck. (forthcoming). Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering.*

G. S. Mann. 2001. A statistical method for short answer extraction. In *Workshop on Open-Domain Question Answering*, pages 23–30.

G. Miller. 1990. Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–312.

J. Prager, D. Radev, E. Brown, A. Coden, and V. Samn. 1999. The Use of Predictive Annotation for Question Answering in TREC8. *Proceedings of the Ninth Text Retrieval Conference.*

TREC-8 Proceedings. 1999. *Proceedings of the Eighth Text Retrieval Conference.* National Institute of Standards and Technology, Special Publication 500-246, Gaithersburg, MD.

TriviaMachine Inc. 1999. TriviaSpot.com. *www.triviaspot.com.*

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.