A Statistical Method for Short Answer Extraction

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

Abstract

This paper presents a simple, general method for using the Mutual Information (MI) statistic trained on unannotated trivia questions to estimate question class/semantic tag correlation. This MI method and a variety of question classifiers and semantic taggers are used to build short-answer extractors that show improvement over a hand-built match module using a similar question classifier and semantic tagger.

1 Introduction

Many of the recent question answering systems integrate statistical NLP/IR tools with a hand-crafted component, a *question class/semantic tag (QC/ST) match* module (Prager et al., 1999), (Breck et al., 1999). Hovy et al. (2001) describes a parsing method for learning QC/ST match. Ittycheriah et al. (2001) trains on trivia questions annotated with the semantic tag of the answer to build a Maximum Entropy model which predicts semantic tags given a question. When the Max-Ent model is used, the estimated probabilities are thrown out and only the most likely tag is returned.

This paper presents a novel method for learning QC/ST correlation from unannotated data. The method introduced is based on the Mutual Information (MI) statistic (Section 2) and is trained on a trivia question database (Section 3) using a question classifier and semantic tagger.

The MI method is general in that it can be applied to a wide variety of question classifiers and semantic taggers. In this paper we examine a few different methods questions classifiers and semantic taggers described in Figure 1.

This MI QC/ST match module, along with a question classifier and semantic tagger, can function as a *short answer extractor* that selects a short answer from a sentence given a question.

Question Classifiers:

1. U : a simple initial unigram model
2. UH : a slightly more complicated model that combines initial unigram and wh-phrase heads
3. Qgrok : a hand-built question typing mechanism (Breck et al., 1999)

Semantic Taggers:

1. NE : Phrag, a HMM Named Entity Tagger (Breck et al., 1999)
2. WN : WordNet (Miller, 1990)

Figure 1: Question Classifiers and Semantic Taggers Investigated

To simplify the problem, we make the assumption that all answers are strictly one word in length[1]. Even so, this task is non-trivial and relevant especially in the case of trivia questions where most of the answers are only one or two words long. The disparity of performance in the 50-byte and 250-byte TREC-8 Question Answering evaluations (Voorhees, 1999) gives further evidence that extracting a shorter, multi-word answer from a longer, sentence length answer is a task worthy of consideration in its own right. We empirically test the performance of the short answer extraction on a held-out set of trivia questions and compare with a number of baseline systems including a hand-crafted system that uses a similar question classifier and semantic tagger (Section 4).

2 MI as an Estimator for Question Class/Semantic Tag Correlation

A simple approach to building a question answering system would be to (1) collect a huge database of questions and answers, and (2) when a question is asked, look for it in the database and return the

[1]Recent experiments have used a Base NP tagger to extract full phrases given the best one-word answer but a thorough evaluation has not been completed.

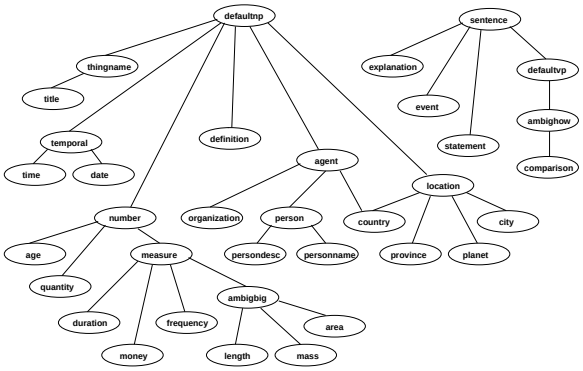

Figure 2: Qgrok's Hand-Built Question Class Hierarchy

answer if it exists. However, the probability of a question repeating is small[2].

Question classifiers and semantic taggers are methods for generalization so that a system can make reasonable guesses on unseen questions. Most question classifiers either scan the question for lexical keywords or extract information from syntactic parses of the question (Harabagiu et al., 2000) to derive the appropriate class. The semantic tags are typically generated by running a MUC-style (Chinchor et al., 1999) Named Entity tagger over the data. Exactly how to use these generalizations is the topic of this paper.

2.1 Limitations of Hand-Crafted Approaches

Most prior approaches to using QC/ST matches to answer questions are based on manually designed hierachies for question classes and semantic tags (e.g. Breck et al. 1999) (see Figure 2). In these systems a hand tuned function determines the match between nodes in the question class hierarchy and the semantic tag hierarchy.

This simple matching is problematic. Both question classifiers and semantic taggers are noisy and error-prone. While there is typically a most frequent semantic tag corresponding to a question class, often a number of other semantic tags are legitimate. For example, "Who" questions are most frequently answered by People, but can also be answered by Organizations ("The NSA"), Locations ("China"), and sometimes miscellaneous proper names ("The New York Times") through metonymic coercions. However, the prior likelihood of any of these semantic tags being the tag of an answer to a "Who" question might be different so they should not be treated as equally good indicators.

A deterministic matcher which makes a hard yes/no decision is likely to reject many good answers. Additionally, the process of building the matcher by hand means that every time the question classifier or semantic tagger changes, the matcher has to be rebuilt and optimized.

2.2 The MI Model

Given training data, even if it is unannotated, we have other options than hand-crafting QC/ST hierarchies and match functions. The model presented here learns statistical correlations for QC/ST matches and thus provides a graceful way of compensating for problems created by noise and variability in the semantic tag of the answer. The task for this model is to pick out the word in a sentence that is the most likely answer to a question, solely based on the question class. Formally, given a question (Q) with a unique question class (QC) and an explanation (E) consisting of a set of words (W), each W with a distribution over semantic tags (ST), the model picks

$$\hat{W} = \underset{W \in E}{\operatorname{argmax}}\ \phi(Q, W) \quad (1)$$

$$\cong \underset{W \in E}{\operatorname{argmax}}\ \phi(QC, W) \quad (2)$$

as the answer. The (pointwise) Mutual Information (MI) statistic is a reasonable candidate for the desired function, since it computes the degree to which two events occur together.

$$\mathrm{MI}(a, b) = \frac{Pr(a, b)}{Pr(a)Pr(b)}$$

With the application of the chain rule and assumption of the independence of the question class and answer given the semantic tag, the following manipulation shows a way to decompose the raw MI statistic into available probabilities.

$$\mathrm{MI}(\mathrm{Q}, \mathrm{W}) \cong \mathrm{MI}(\mathrm{QC}, \mathrm{W}) \quad (3)$$

$$= \frac{Pr(\mathrm{W}, \mathrm{QC})}{Pr(\mathrm{W})Pr(\mathrm{QC})} \quad (4)$$

$$= \frac{Pr(\mathrm{W}|\mathrm{QC})}{Pr(\mathrm{W})} \quad (5)$$

$$= \sum_{ST} \frac{Pr(\mathrm{W}|\mathrm{ST})Pr(\mathrm{ST}|\mathrm{QC})}{Pr(\mathrm{W})} \quad (6)$$

$$= \sum_{ST} \frac{Pr(\mathrm{ST}|\mathrm{W})Pr(\mathrm{ST}|\mathrm{QC})}{Pr(\mathrm{ST})} \quad (7)$$

$$= \sum_{ST} Pr(\mathrm{ST}|\mathrm{W})\mathrm{MI}(\mathrm{QC}, \mathrm{ST}) \quad (8)$$

[2]Though not insignificant as services like AskJeeves have shown.

This derivation shows that if we compute (8) we approximate MI(Q,W) given the independence assumptions above. In effect, we have learned how predictive a QC/ST pair is. Table 1 gives an example of the type of information learned by the MI model.

2.3 Estimating the MI Model from Unannotated Data

Estimating the above probabilities can be done with a trivia database that contains a large number of questions and answers. The method is the following:

- For each question identify the question class.
- Apply the semantic tagger to the trivia database to generate $Pr(ST|W)$. Alternatively, tag a very large corpus to generate high precision priors, and ensure that answers in the training data are tagged at least once. If high quality priors are available, as outputs from an HMM for example, they also might be able to be used as fractional counts to estimate the probabilities[3].
- Estimate

$$Pr(ST|QC) = \sum_{W} Pr(ST|W)Pr(W|QC)$$

As stated above, this estimation method makes the assumption, expressed in equation (3), that for each question there will only be one question class. For most current Q/A systems this is the case. Perhaps future systems will have more sophisticated question classifiers that assign a probability to a number of question classes. To accomodate increased sophistication, these formulas will change slightly, but the general method may still be applicable.

When this MI method is trained by the above method, it takes into account the actual performance of the semantic tagger on data. Ittycheriah et al. (2001) builds a statistical model on annotated data which predicts semantic tag from the question and notes that improvement in this prediction does not necessarily lead to higher performance, since there is a complex interaction between this module and the semantic tagger and answer selector. One advantage of training on unannotated text using the MI approach is that the correlation between the question class and the performance of the semantic tagger is explicitly modeled.

[3]In this paper, even though Phrag is a HMM, its priors were unavailable, so it was treated as a non-statistical tagger

Q Class	Sem Tag	MI(QC,ST)
In Country	Country	39.605
	Location	3.683
	City	3.562
	Organization	1.572
	Name	1.116
	Person	0.175
	Other	0.012
	Date	0.028
	Noun Group	0.005
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: Examples of MI(QC,ST) induced from the Phishy (PH) trivia database

3 Trivia Questions

With the advent of the Internet, trivia games are becoming big business. The general public submits questions, and trivia game companies award prizes to those who correctly answer those questions. Some of these trivia databases are quite large, reaching nearly two hundred thousand trivia questions (Ford).

In this paper we use two trivia databases as main resources : "Phishy" or PH (MacDonald) and "Triviaspot" or TS (Trivia Machine Inc.). PH has approximately 5k questions, each with the correct answer. TS is larger, but only a small part (11k questions) is currently available to us for 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, in fact, justifications for the answer (as in Figure 3). Others provide additional information for those interested in the topic of the question (e.g. "Leonardo Da Vinci described ideas for contact lenses in 1508.") or for those upset at answering wrong (e.g. "Franklin wore glasses, but didn't invent them."). Both

	TREC-8	TREC-9	CBC	PH	TS
#	200	693	651	4,857	10,959
What	**.299**	**.433**	**.235**	**.292**	**.334**
Who	**.234**	**.162**	**.147**	**.109**	**.202**
How	**.154**	**.075**	**.180**	**.047**	**.029**
Where	**.104**	**.101**	**.118**	.018	.026
When	**.095**	**.069**	.101	.001	.006
In	.025	.004	.012	**.097**	**.057**
Which	.045	.004	.014	**.314**	**.145**
Why	.010	.003	**.141**	.002	.003
total	.970	.876	.945	.881	.802
(other)	.030	.124	.125	.119	.198

Table 2: Distiributions of initial unigrams for questions in five collections, with the five most common for each dataset in **bold**

Question: Who invented eyeglasses?
Answer: Chinese
Wrong: Benjamin Franklin
Wrong: Thomas Jefferson
Wrong: Japanese
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.

Figure 3: TS Trivia Database Entry #42764

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.

These trivia questions provide an appealing source of questions for those interested in question answering. First, they cover a wide domain of knowledge and demonstrate complex grammatical constructions. For example, the distributions of initial unigrams for questions in the TREC-{8,9} Question Answering evaluation, the CBC reading comprehension data set (Breck et al., 2001), and the trivia question database are quite different. Figure 2 shows distributions for the union of the top five most common initial unigrams for five question collections. The major differences between the trivia questions (TS and PH) and the other collections are the percentage of outlying question initial unigrams, the increased percentage of "Which" and "In" questions, and the decreased number of "When" and "Where" questions.

Second, the questions typically ask for simple facts whose answer is often only a word or two. The average answer length, excluding stop words, is 1.36 for PH and 1.79 for TS. TREC-{8,9} answers were similar in length.

Finally, trivia question databases encode a tremendous amount of factual information. In this paper we demonstrate how to extract a certain type of information from this heterogeneous database: question class and semantic tag correlation. However, there is other loosely structured information in these databases. For example, if the question begins "In which county" and the answer is "Indonesia", a system could learn that Indonesia is a country. Preliminary experiments using this method indicate that even a small trivia database (PH w/5K questions) has enough instances of this specific construction to contribute new proper names to the WordNet hierarchy. It seems likely that there exists more information of this type within trivia question databases waiting to be discovered.

3.1 Collecting Explanations from the Web

While some trivia databases include explanations, not all do. In these cases, it might be possible to test performance of a short answer extractor if explanations (sentences that might answer the question) could be found. Since the PH database included no explanations, we tried one method, where we searched the Internet using the Google search engine and the Q/A words as the query. Sentences with the highest overlap with the questions were chosen and then sentences without any of the short answer terms were eliminated.

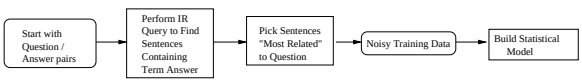

Figure 4: How to Find Possible Sentence-Length Answers on the Web

As a result of this search procedure, we collected nearly 22k explanations for approximately

3k 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 and many were odd mixtures of sentence fragments and random punctuation. Since the explanations were collected automatically, in some cases we found multiple explanations for the same question. The results reported for PH in the next section report scores computed over all question/explanations pairs.

We tested the efficacy of the MI model by examining the performance of this model at selecting a one-word answer from the web explanations for PH and from the explanations provided by TS.

Our experimental method for testing the statistical method was as follows:

1. Divide the questions into a testing and training set (Table 3). Tokenize all explanations using a text chunker (Florian and Ngai 2001) and ensure that those in the test set contains an answer.
2. Run the question classifier and semantic tagger over the training set. **NE-U** refers to the first MI system tested, where the semantic tagger = Phrag, and question classifier = Initial Unigram. In training, throw out unigrams that don't occur more then ten times in the data. Estimate a back-off "all" as

$$\frac{\sum_Q \sum_{W \in \text{correct}} P(ST|W)P(W|Q)}{\sum_{Q,E} \sum_W P(ST|W)P(W|Q)}$$

   Note that the numerator is $P(ST|\text{answer})$, and the denominator is $P(ST)$.
3. Estimate $\text{MI}(ST, QC)$ from the training set as described in Section 2.
4. For each question in the test set run the question classifier to determine the question class, backing off to "all" in the case of unseen unigrams. Use the semantic tagger to assign a distribution $Pr(ST|W)$ to each word in the explanation. Rank all non-stop words in the sentence according to :

$$\text{MI}(Q, W) = \sum_{\text{ST}} \text{MI(ST, QC)} Pr(\text{ST|W})$$

   This process is depicted in Figure 5 below.
5. Evaluate using two methods :
   (a) correctness of the top ranked answer (**correct**)
   (b) the reciprocal answer rank (**rar**) as used in TREC, (1/rank of first correct answer) for the top 5 answers

	#training questions	#testing questions	avg ans length
PH	3,105	1,714	1.36
TS	6,681	4,275	1.79

Table 3: Statistics of Trivia Databases PH & TS

We compared the performance of this system with a number of baselines. The most naive method was **Random** which is the expected performance of a system that picks a word at random within the sentence (excluding stop words). We also tried **Word Order**, which ranked words by their position in the sentence (i.e. first word ranked first).

4 Experimental Results

Vanilla used Qgrok and Phrag as question classifier and semantic tagger respectively and a handcrafted match module to detect matches. Qgrok is only slightly more complicated than the initial unigram method. **NE-U** is the first MI model, which uses Phrag as a semantic tagger and initial unigrams as a method of question classification.

	PH		TS	
	Correct	Rar	Correct	Rar
Random	.108		.170	
Word Order	.150	.272	.440	.548
Vanilla	.163	.282	.350	.480
NE-U	.315	.478	.414	.571

Table 4: Peformance of Baselines vs. NE-U

Table 4 shows the results of this experiment. The improvement in both absolute correct and in absolute reciprocal answer rank achieved by NE over the baseline Vanilla system is surprising. Both systems use very similar question and semantic taggers, with the largest difference being that NE-U uses a $Pr(ST|W)$ distribution, while Vanilla chooses $\text{argmax}_{ST} Pr(ST|W)$ as computed by Phrag. The question classification mechanism used in Vanilla is more complex than that used in this system (though this might be a source of problems – see Section 4.2).

Another striking result from these experiments is the impressive performance that is achieved by using Word Order on the TS database. The fact that this method can do so well illustrates that the explanations in TS are unnatural – most of them start with the correct answer. This information was not used in the rest of the paper or in any of the models, since it is a property of this specific database of questions, as the results on PH demonstrate.

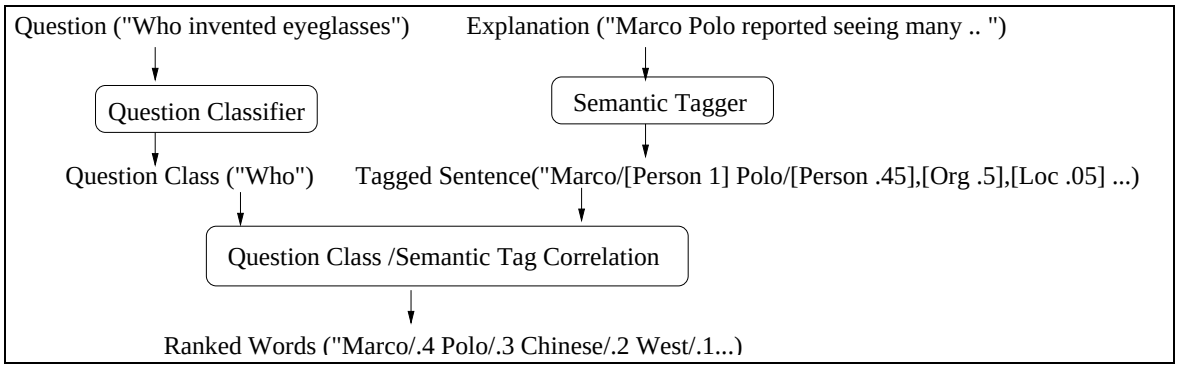

Figure 5: The Short Answer Extractor : Dataflow in the NE-U system

4.1 Integrating WordNet

Many trivia questions are not answered by proper names or Named Entities for that matter. However, all hope is not lost for trying to match these questions with answers, since common nouns can be semantically tagged. For example, there are many questions that ask "Which color" or "Which animal". These types of questions suggest that a noun hyponym hierarchy might be useful in learning the correct semantic tag for these questions. Even a rough estimate for common nouns most likely to answer "What" questions might give a useful bias.

WordNet (Miller, 1990) contains a large coverage common noun hyponym hierarchy and seems an obvious first choice for any kind of generalization along these lines. WordNet also includes a sizable proper noun hierarchy though it achieves much less coverage than contemporary Named Entity taggers.

WordNet has been previously utilized in Q/A systems (notably Harabagiu et al. (2000)) . However, prior usage has relied on hand-coding relationships between questions and semantic tags. In this section we demonstrate that the MI criterion used in Section 2 can also be used to generalize over WordNet classes, and that combining these generalizations can improve the performance of the short answer extractor.

To integrate WordNet classes, we used the same model as Resnik (1995). Training on the Brown Corpus (Francis and Kucera, 1982), we counted an appearance of a WordNet class (WC) every time one of its children appeared.

$$\text{freq}(WC) = \sum_{n \in \text{ children}(WC)} \text{freq}(n)$$

and then used these counts to estimate

$$Pr(WC) = \frac{\text{freq}(WC)}{N}$$

where N is the number of words in the corpus. We used a similar method to estimate $Pr(WC|Q)$ and for $Pr(WC|W)$ gave a uniform distribution over all classes it was a member of (all ancestors of all senses).

We tested this system's performance alone **(WN)**, then on a system which mixed the correlation from Named Entity semantic tags and from WordNet tags with a parameter λ **(CB)**.

$$\text{MI}_{CB}(Q,W) = (1-\lambda)\text{MI}_{NE}(Q,W) + \lambda \text{MI}_{WN}(Q,W)$$

We used $\lambda = .1$ for for PH and $\lambda = .01$ for TS. Table 5 presents the results of testing this system. While using WordNet didn't result in a large improvement for these methods, the combined CB achieved the best performance overall, when the parameters were slightly tuned. Ongoing work looks at whether $P(WC|Q)$ and $P(WC|W)$ can be better estimated and combined more effectively with other sources of data.

	PH		TS	
	Correct	Rar	Correct	Rar
Vanilla	.163	.282	.350	.480
NE-U	.315	.478	.414	.571
WN-U	.246	.396	.262	.409
CB-U	.321	.485	.424	.579

Table 5: Performance of different Semantic Taggers, with Q Class=Initial Unigram

4.2 Variation in Question Classifiers

We compared the performance of different question class modules using the baseline, NE, and the best performer, CB, as semantic taggers. The first two question class methods have been described earlier : **U** initial unigrams and **Qgrok** which was used for the Vanilla system.

We also tried a third method of question typing (**UH**) which was slightly more complicated.

It used not only the initial sentence unigram, but also the head word of the initial wh-phrase (when it could find one), again modeling these types only when they occured more than ten times in the training corpus. As a result of this process, it's possible to create a question class that hasn't appeared yet in data. In these cases, the model backed-off to the initial unigram or to "all". This back-off was not optimal, and future investigations into more effective methods may be profitable.

The results from these comparisons are listed in Table 6. The performance on each of the test sets is strikingly different. On PH, the Unigram+Head (UH) method achieves the best performance, while on TS Qgrok does. One explanation behind these differences might be found in Table 2, which shows that PH has a much higher concentration in fewer initial unigrams (.881) which TS is more varied (.802). This might explain why Qgrok, which does more complex question classification achieves a bigger benefit in TS than UH. These results suggest that while simple question classification works reasonably well for simply phrased questions, it degrades with more complicated phrasing.

	PH		TS	
	NE	CB	NE	CB
Qgrok	.292	.299	.427	.433
Unigram	.315	.321	.414	.424
Unigram + Head	.328	.345	.408	.418

Table 6: Effect of Different Question Classifiers with Respect to Overall Performance

4.3 Using Wrong Answers

One final piece of information in some trivia question databases is a set of wrong answers for each question. In multiple choice trivia questions, typically the correct answers and incorrect answers all could be possible responses to the question or else a contestant would be able to answer the question without any other knowledge. We assume that one of the main similarities is that all answers have a semantic tag that is highly correlated with the question. In other words, the tag of the wrong answers should be a possible tag expected by the question, and thus should help estimation of the question class/semantic (QC/ST) tag correlation.

We tested this hypothesis by using these wrong answers to estimate MI(ST,QC), and seeing if indeed they improve performance. For these experiments, we took the training questions, correct answers (C) and incorrect answers (I) and recomputed $Pr(ST|QC)$ using :

$$Pr(ST|QC) = \sum_{W \in \{C \cup I\}} Pr(ST|W)Pr(W|QC)$$

For these experiments we used only the TS database, since PH did not contain wrong answers. Table 7 shows an improvement in performance with the use of wrong answers in estimating MI(ST,QC), though not a large one.

QC	ST	plain	+wrong
Unigram	NE	.414	.424
	CB	.423	.433
Unigram + Head	NE	.408	.413
	CB	.418	.423

Table 7: Effect of Using Wrong Answers As Additional Training (TS)

4.4 Probing the Open Domain Nature of Trivia Questions

One important question is to what degree models learned from trivia question databases can be broadly applied to question answering in general. To begin an answer to this question, we examine how well models learned on one trivia database can be used on another unrelated one. We built models from PH and TS, exchanged $Pr(ST|QC)$ models, and re-tested. With this replacement, we expect a degradation solely due to QC/ST correlation differences. We did not replace the other two models since if one was given a new set of questions and a corpus of sentences, $Pr(ST|W)$ and $Pr(ST)$ could be calculated for that domain without knowing the correlation between questions and answers. Formally, we defined Pr^{PH} to be probailities estimated from the PH corpus, and Pr^{TS} to be those estimated on TS. The model we normally test on PH is built by:

$$\text{MI}^{PH}(Q,W) = \sum_{ST} \frac{Pr^{PH}(ST|W)Pr^{PH}(ST|QC)}{Pr^{PH}(ST)}$$

Instead in this model, we computed

$$\text{MI-X}^{PH}(Q,W) = \sum_{ST} \frac{Pr^{PH}(ST|W)\boxed{Pr^{TS}(ST|QC)}}{Pr^{PH}(ST)}$$

The performance detailed in Table 8 shows that the degradation is minimal when models are shared across two trivia databases. This result suggests either that the trivia questions, at least to a first approximation, are very similar or that what these models have learned is a general phenomena of question answering. Which hypothesis is correct is left to future research.

	PH		TS	
	cor	rar	cor	rar
MI	.315	.478	.414	.571
MI-X	.321	.461	.386	.549

Table 8: Performance when $Pr(ST|QC)$ is exchanged, using Unigram Question Classes, and NE Semantic Tagging

5 Conclusions

In this paper we examined a component of Q/A systems which is often over-looked : the component which measures fit between question classes and semantic tags. We described a novel way to use the mutual information statistic and an unannotated corpus to automatically induce correlations between semantic tags and question classes. We have shown that wrong answers can help improve performance and that different semantic taggers can be combined to improve performance. The MI statistic as presented here can be used as "glue" to combine a variety of question class/semantic tag components, and as such it is of general usefulness to the Q/A community.

The similarity between trivia databases, shown by cross-training experiments, is another interesting result. Although it does not prove that trivia questions constitute an open domain, it suggests that trivia questions are at least a self-consistent domain.

We have shown that another component of a Q/A system can be built statistically to yield nice performance when even simple statistics are used. This prompts the question : what other components can be built statistically? Already some components are consistently built by statistical methods (semantic taggers, information retrieval engines), yet some remain predominately handcrafted (e.g. question classifiers – with Ittycheriah et al. (2001) as an exception), and thus are prime targets for statistical methods.

This paper demonstrated performance on extracting one-word answers, but this method can be extended to extracting multiple words. We built a system which chooses the base noun phrase containing the highest ranked word, but have not completed evaluation.

Aside from its use in question answering, once the short answer extraction task can be performed with high precision, systems will be able to extract facts from heterogenous text. This will be an enabling technology which will allow integration with symbolic processing systems to allow for complex natural language understanding.

Acknowledgements

The author would like to thank Ellen Riloff, John Hale and Jun Wu for their insightful and helpful comments and suggestions and John Burger, Marc Light, Eric Breck, Inderjeet Mani, and Lynette Hirshman for providing Phrag, Qgrok and the various question corpera. Thanks are also due to Grant MacDonald, and to Tiffany Kosel, Hazel Kight, and Trivia Machine Incorporated for the trivia questions used in these experiments.

References

E. Breck, J. Burger, L. Ferro, D. House, M. Light, and I. Manni. 1999. A sys called qanda. *Proc. of TREC-8.*

E. Breck, M. Light, G. Mann, E. Riloff, B. Brown, P. Anand, M. Rooth, and M. Thelen. 2001. Looking under the hood: Tools for diagnosing your question answering engine. *ACL Workshop on Open-Domain Question Answering.*

N. Chinchor, E. Brown, L. Ferro, and P. Robinson. 1999. 1999 named entity recognition task definition. *Tech Report.*

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

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

R. Florian and G. Ngai. 2001. Multidimensional transformation-based learning. *Conference on Natural Language Learning.*

W. Francis and H. Kucera. 1982. *Frequency Analysis of English Usage.* Houghton Mifflin, Boston, MA.

S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Mor. 2000. Falcon : Boosting knowledge for answer engines. *Proc. of TREC-9.*

E. Hovy, L. Gerber, U. Hermjakob, C-Y. Lin, and D. Ravichandran. 2001. Towards semantics-based answer pinpointing. *Human Language Technologies 2001.*

A. Ittycheriah, M. Franz, W. Zhu, and A. Ratnaparkhi. 2001. Question answering using maximum entropy components. *Proc. of NAACL.*

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

J. Prager, D. Radev, E. Brown, A. Coden, and V. Samn. 1999. The use of predictive annotation for question answering in trec8. *Proc. of TREC-8.*

P. Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of IJCAI.*

TriviaMachine Inc. Triviaspot.com.

E. Voorhees. 1999. The trec-8 question answering track report. *Proc. of TREC-8.*