CS224N
Spring 2010

High-Performance Question Classification Using Semantic Features

Olalere Williams
Stanford University
lere@cs.stanford.edu

Abstract

A question classification system uses machine learning techniques to classify a question by the type of answer it requires. Recent systems have successfully used syntactic parsing to identify small, highly-informative phrases or words within a question to use as the principle features of a probabilistic model. In this paper, I explore the effectiveness of an alternative feature set that makes use of only semantic features. My system achieves a classification accuracy of 86.6% on the standard UIUC benchmark test set (Li and Roth, 2002), making it superior to all but the most recent classification system (Huang et al., 2008).

1 Introduction

In a seminal paper in this area of natural language processing, Li and Roth (2002) used a diverse feature set (consisting of both syntactic features such as POS tags and semantic features such as named-entities) to achieve the baseline performance of 84.2% on a dataset they assembled that has now become standard. In addition to the UIUC data set that they published, they also defined a common classification taxonomy of 50 fine classes and 6 coarse classes that systems could target. Reporting system classification accuracy against the set of fine classes has become the standard evaluation metric in this area.

Krishnan et al. (2005) more recently contributed the concept of an **informer**, a short (usually one to three word) contiguous phrase within the question that can be used to accurately classify it. They adopted a meta-learning framework in which they first trained a sequence

model to classify informers, and then combined features of the predicted informer with more general features into a single large feature vector that was fed into an linear support vector machine (SVM) to classify the overall question. Informers were identified using features derived from a parse of the input question. This approach achieved an overall question classification accuracy of 86.2% (while identifying informers with 85% accuracy). Finally, Huang et al. (2008) derived features from the head-words of the principal noun phrases in a question (such as WordNet (Fellbaum, 1998) hypernyms, which were first introduced as promising features by Krishnan et al., 2005). This approach was seen as a refinement of the concept of an informer (which occasionally included too many words, thus polluting the combined feature vector with misleading features). At the time of writing, this system achieves the highest published performance of 89.2% question classification accuracy.

In this paper, I explore an alternative feature set that is based purely on semantic features. The inspiration for this strategy comes from Li and Roth's (2002) observation that semantic features provided the greatest benefit for the overall classification system. Initially I set out to investigate to what extent semantic features could be used in isolation to create a question classifier with comparable performance to the baseline established by Li and Roth (2002). As it turns out, combining a rich set of semantic features with Krishnan et al.'s (2005) notion of an informer results in a system that is second only to Huang et al.'s (2008) most recent system.

2 Methodology

My overall system consists of a maximum entropy Markov model used as an informer tagger, and a maximum entropy classifier used to classify overall questions based on a large feature vector which combines informer features with features of all words in the question (informer words or otherwise).

2.1 Informer tagger

Following Krishnan et al. (2005), I build sequence model to indentify informers as an intermediary step to classifying the overall question. I chose to pursue this method because I thought a reasonable classification of informers could be achieved using only semantic features, whereas the more successful approach of Huang et al. (2008) seems to necessitate a syntactic parse of the input question. Training data for the informer tagger was obtained by hand-labelling 2000 questions from the standard training set. Similarly, for test data I hand-labelled the 500 questions that form the standard test set for the question classification problem.

My feature set for the informer tagger is compact and effective. For each word in the question (i.e. for each member of the sequence of words that will eventually be tagged with one of three possible labels – O1 for tokens that come before the informer, I for informer tokens and O2 for tokens that come after the informer), I use as basic features: (1) the word itself, (2) the words immediately preceding and following it (if they exist), (3) whether or not the previous word was a "question word" (one of "what", "which" or "how"), (4) the previous label and (5) whether or not the question as a whole consists of six tokens or less. I then also model various interactions between these features by hand (as is required by a maximum entropy Markov model).

All of these features are highly intuitive, except perhaps the last feature. I included this after realizing that definitional questions occur frequently in the training and test sets, and that this class of question can easily be identified by the relative brevity of those questions. Moreover, I think that it is highly plausible that real question answering systems would also face a large number of short, definitional questions. This means that far from being an *ad hoc* feature that exploits idiosyncrasies of the training and test sets, I believe this feature accurately models an important class of question.

The incremental gains secured by each feature are shown in Table 1. I follow Krishnan et al. (2005) in reporting Jaccard overlap as the principal measure of informer tagger performance (see that paper for justification), but I also include more common evaluation metrics. Unfortunately, however, they did not make available the dataset they used for training and testing the informer tagger. Since our datasets were annotated differently, the numbers here are incommensurable to Krishnan et al.'s (2005) figures.

Feature Set	Jaccard overlap	Precision	Recall	F1
Current word	0.380	0.840	0.410	0.551
+ Previous label	0.441	0.861	0.475	0.612
+ Previous word	0.473	0.861	0.512	0.642
+ Question word	0.494	0.889	0.526	0.661
+ Brevity(6)	0.748	0.882	0.831	0.856
Brevity(5)	0.767	0.910	0.831	0.868
Brevity(7)	0.564	0.826	0.640	0.721
+ Next word	**0.806**	**0.904**	**0.882**	**0.893**

Table 1: Incremental performance gains (on informer tagging) by feature.

As you can see from Table 1, the brevity feature is crucial to the performance of the informer tagger, and that using a token length of 6 provides the greatest benefit (alternative parameters for the brevity feature are shown in rows that do not start with a "+").

Figures shown above are those obtained from training on a training set of 2000 labelled examples. I initially developed the system on a training set of 1000 examples, but was disappointed by the performance of the tagger. Table 2 shows the significant impact (approximately 5% improvement in Jaccard overlap) of increasing training data. While it is likely that I would have experienced diminishing returns on using more training data, this would likely still have been helpful up to a point (I do not use more training data simply for lack of time to annotate the data).

Training Examples	Jaccard overlap	Precision	Recall	F1
1000	0.757	0.873	0.851	0.862
2000	0.806	0.904	0.882	0.893

Table 2: Impact of size of training set on performance of final informer tagger.

2.2 Combined feature vector

In order to classify overall questions, I use the informer tagger to predict an informer for a given question, and then extract a set of informer features and a set of word features from all the words in the question (informer or not). These two sets of features are then combined into single feature vector that is fed to the maximum entropy classifier.

2.2.1 Informer features

I include all n-grams of the informer as separate features. In addition, for every noun in the informer, I include all WordNet hypernyms of all senses of the noun (using Finlayson's (2007) WordNet interface). These features follow Krishnan et al (2005) directly. Briefly, the intuition is that although for a question such as *What was the name of the author of Macbeth?* the correct informer is "author", this word is significantly narrower than the class to which the question belongs (defined as HUM:ind according to the standard taxonomy; a class that encompasses all humans). Hence, using hypernyms of the informer words as features is likely to produce a broader word (such as "person") that is more reliably correlated with the correct class.

2.2.2 Word features

Here I again follow Krishnan et al. (2005) in many respects, but also make some important additions to their feature space. Like them, I found that using n-grams of the question were an effective feature (but contrary to them I found unigrams and not bigrams to provide the greatest performance gain). An addition I make is that I reintroduce the named-entities feature that was present in the original feature set proposed by Li and Roth (2002). Named-entities (one of three classes: person, location or organization) are identified here using the Stanford NER package (Finkel et al., 2005).

3 Results

The incremental impact of the various features described above (and others used during development of the final system) is shown in Table 3.

Feature Set	Coarse Accuracy	Fine Accuracy
Informer n-grams	0.888	0.804
+ informer hypernyms	0.898	0.822
+ unigrams	0.916	0.860
bigrams	0.914	0.852
unigrams, unigram hypernyms	0.912	0.846
unigrams, unigram hypernyms, bigrams	0.922	0.854
+ named-entities	**0.910**	**0.866**

Table 2: Incremental performance gains (on question classification) by feature.

[The last row of the above table shows the results of the final system (with features: informer n-grams, informer hypernyms, unigrams and named-entities). Each row shows an incremental impact of the listed feature, where the baseline performance is given in the first row the table.

Rows that do not begin with a "+" show the impact of various alternatives at that level of the feature composition. The last row then shows peak performance with the first listed alternative (the unigrams feature); that which produces the best overall performance.]

It is interesting to note that my results seem to favour simplicity; a small number of effective features. I take this to reflect the problem of sparsity in expansive feature sets. For example, although we get local improvement from features such as unigram hypernyms and bigrams, these do not combine with named entities to produce the best overall system (although only figures for the most effect system are reported above). This suggests that the extra features may simply take probability mass away from features that are prominent in the classification decision.

Remarkably then, I achieve performance slightly superior to Krishnan et al. (2005), using mostly features that they themselves propose (with the noted exception of named-entities which provides the final performance boost). However, line three of Table 2 shows that, even without named-entities (i.e. using solely features suggested by Krishnan et al., 2005), my system is only 0.2% worse than their system. This suggests that informer tagging can be effectively accomplished without the use of syntactic parsing, because the overall systems perform similarly when using the same combined feature vector but tagging informers differently (granted, however, that I have modelled interactions between the various features by hand, and Krishan et al. do not report how they dealt with feature interactions).

This final system would likely benefit from better informer tagging (which would in turn improve informer features). Table 4 shows the difference of performance of the final system when the informer is trained on 1000 examples rather than the 2000 examples that was used in reporting the figures above (we see a 2.4% increase in performance on the fine classification problem). I suspect that had I been able to label all 5500 examples from the UIUC data set for informer training (as did Krishnan et al., 2005), my overall system would have benefited.

Training Examples	Coarse Accuracy	Fine Accuracy
1000	0.914	0.842
2000	0.910	0.866

Table 4: Impact of size of informer training set on overall question classification system.

4 Conclusion

As I have demonstrated, semantic information alone (i.e. without information derived

from syntactic parses of questions) is sufficient to produce a high-performance question classification system. The successful strategy that I employ involves first identifying informers, short phrases that are well-correlated with question class, and then using features extracted from these informers and from words in the question to produce a global feature vector that can then be used to classify overall questions. Identifying amed-entities within the question is a strong feature that enhances the performance of the overall system (beyond what it achieves given the other features proposed by Krishnan et al., 2005). The final system that I propose is competitive with existing systems, achieving a question classification accuracy of 86.6% on the standard UIUC data set.

References

C. Fellbaum. 1998. An Electronic Lexical Database. The MIT press.

J.R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005),* pp. 363-370.

M. Finlayson. The MIT Java WordNet Interface. 2007. Software available at: http://projects.csail.mit.edu/jwi/

Z. Huang, M. Thint and Z. Qin. 2008. Question Classification Using Head Words and their Hypernyms. *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pp. 927-936.

V. Krishnan, S. Das, and S. Chakrabarti. 2005. Enhanced Answer Type Inference from Questions using Sequential Models. *The conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

X. Li and D. Roth. 2002. Learning Question Classifiers. *The 19th international conference on computational linguistics*, vol. 1, pp. 1–7.