CS674 Project Report: Answer-based Question Classification

Sumeet Agarwal (Y2387), David Doukhan (EXY502)
Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur, India - 208016
{sagarwal, david}@cse.iitk.ac.in
Guide: Dr. Amitabha Mukerjee

November 20, 2005

Abstract

Question-Answering systems are being widely studied for their range of potential applications, in particular in Web Information Retrieval. Here, we describe a new idea for automatically classifying questions based on the type of answer expected, thus making it easier to search for the relevant information. We also look at some experiments done by us based on our method, and present the results obtained therein. Finally, we try and analyze our work to see what improvements can be made in order to obtain better classification accuracy.

1 Introduction

The basic goal of a Question-Answering system (see, for example, [6]) is to extract answers to open-domain questions from a given set of content documents. There are two types of approaches which are employed for this purpose: one is based on matching surface forms, and is known as the "bag of words" method; the other looks to go deeper and analyse the questions and documents semantically, thus trying to "understand" what is being asked. Our work is more in line with the second idea, as we look to use semantic features to classify questions. The goal is to make it easier to pinpoint the answer, by restricting our search to information of a certain kind. Previous work on question classification [7] has been based on the idea of manually annotating a training set of questions with category labels, and then using this to learn the classifier, based on certain features extracted from the questions. Here we look to try unsupervised learning, using an unlabeled set of questions and answers. The idea is to first cluster the questions based on some semantic features, and then assign a label to each cluster by analysing the answers to all the questions in that cluster. Subsequently, we use these labels to construct a labeled dataset, which the classifier is finally trained on.

The rest of this report is organized as follows: Section 2 outlines our entire methodology; Section 3 looks at the process of extracting features from the questions and using them to form clusters; Section 4 describes how the answers are used to come up with cluster labels; Section 5 explains the training and testing procedure, and also presents some experimental results; and Section 6 states the overall conclusions of the work done, as well as directions for future improvements .

2 Methodology

The following steps constitute our method for question classification:

1. We take a training set consisting of questions and answers, obtained from the TREC database (`http://trec.nist.gov`).

2. Each question in the training set is tagged and parsed and relevant semantic features are extracted to build a feature set. Thus, we get a feature space representation of the question set.

3. The Expectation Maximization algorithm [3] is used to cluster the questions on the basis of the extracted features.

4. Corresponding answer clusters are formed, with all the answers to questions in a given cluster being grouped together.

5. Each answer cluster is analyzed in two ways: named entity identification and semantic analysis using WordNet [4]. Based on the information thus obtained, a class label is attached to each question cluster, indicating the "type" of the questions in that cluster. For example, a cluster may be labeled as LOCATION, indicating that the questions are all asking for the name of a place.

6. Using the cluster labels and the features extracted earlier, a labeled dataset of the questions is built. This is used to train a Support Vector Machine [2] classifier, which can then be used to attach labels to new questions.

7. Roth and Li's [7] manually annotated dataset is used to test the classifier. We use questions belonging to categories which are similar to those obtained by our clustering method.

3 Feature Extraction and Clustering

The goal here is to obtain a set of features which are sufficiently rich to determine the question type. However, this task is not easy, since we are looking at unsupervised learning, and so the system has no way of determining to what extent different features are relevant for the kind of classification we are looking for. We focused primarily on semantic features, since the nature of the labeling we are looking to do is semantic.

To start with, the questions were tagged with Part-Of-Speech (POS) tags, using Ratnaparkhi's Maximum Entropy Model-based tagger, MXPOST [8]. They were then parsed using Collins' Parser [1], in order to obtain phrasal chunks (like Noun Phrase, Verb Phrase etc.), and the entire parse tree structure. Having gotten this, the following features were extracted:

- The key question word: one of Who, What, When, Where, Why, Which or How. These words are separately tagged by the POS tagger.

- The phrase tag of the head noun phrase: one of NP-A, NPB or NNP, as determined by Collins' parser. This gives information about the context of the head noun, i.e. the kind of clause it is embedded in. For instance, NPB indicates a base, or non-recursive, noun phrase.

- The semantic category of the head noun, obtained using WordNet [4]. The WordNet ontology allows us to locate each word as a node in a tree, and by going up the tree, we can get more and more general categories to which that word belongs. We used the word found three levels below the root; this allows us to obtain a fairly generic category. For example, for words like city, town, country etc., we get the type *l*ocation.

- The semantic category of the word following the key question word; also obtained using WordNet.

Admittedly, this feature set is rather small; but we were not able to add any further useful features to it. We tried using some other information, such as POS tags, but it only worsened clustering performance. So we used just this feature set, and clustered the questions based on it. For this purpose, we employed the Expectation Maximization (EM) algorithm, as implemented by the Machine Learning toolkit Weka [9]. Since the feature set was not very rich, we expected only coarse clustering to take place; and indeed, only 5 clusters were generated when we ran the procedure on the TREC 2000 and 2001 main task question sets (available online at `http://trec.nist.gov/data.html`). The next task was to label these clusters; the way we did this is described in the next section.

4 Analysing the Answers

Past work in question classification, such as that of Roth and Li [7], has focused on learning a classifier using a manually annotated training set. The key novel element in our approach was the use of known answers to derive question labels for training purposes, so that manual labeling could be avoided. Initially, we tried to employ semantic analysis based on FrameNet to get the labels from the answers, following the approach of Gildea and Jurafsky [5]. However, we found that most of the answers in the TREC database are not focused, since they are extracted from documents like books and newspaper articles. For example, one might have an answer like: "Boris Becker won his first Wimbledon title in 1985". This could correspond to several different questions, such as: "Who won Wimbledon in 1985?", or "When did Boris Becker win his first Wimbledon title?", or "Which major championship did Boris Becker win in 1985?"; clearly, all of these questions belong in different categories. The first one might be labeled PERSON, the second DATE/TIME and the third

EVENT/ENTITY.

So, a single answer could not be reliably labeled. Thus, we decided to try unsupervised learning, forming clusters of questions and then looking to analyse all the corresponding answers together, so that a dominant pattern could be detected, based on which a category could be assigned. In order to do this, we tried two different approaches: Named Entity Identification, and Semantic Analysis using WordNet.

4.1 Named Entity Identification

A Named Entity is essentially a proper noun; it could be the name of a person, place, object, event, or organization. By detecting all the named entities in a given cluster of answers, and then counting the numbers of the different kinds of entities, one can get an idea as to the appropriate type for the cluster as a whole. In order to tag named entities, we used a tool called LingPipe (`http://alias-i.com/lingpipe`). This tool finds the entities, and labels each of them with a tag like PERSON, LOCATION or ORGANIZATION. Apart from this, we also wrote code to detect dates and other numeric quantities, and these were also treated as named entities.

Once all the named entities in each cluster had been found and tagged, we looked at the relative frequencies of the different types across clusters. Based on these, certain clusters could be broadly labeled; for instance, a cluster dominated by PERSON entities would naturally be assigned the same label as a whole. This approach was useful in some cases, but since the number of different entity types is very limited, it cannot offer a thorough analysis. So we decided to try and employ another approach in parallel, and see if we could use the results of both to come up with better cluster labels.

4.2 Semantic Analysis

The key idea of the semantic analysis we carried out was to use WordNet's ontological representation [4] of each word as a node in a tree. This representation allows us to represent a word such as "red" by the corresponding tree:
entity => abstract entity => abstraction => attribute => property => visual property => color => chromatic color => red

This kind of representation allowed us to use a probabilistic approach: basically, we considered the tree corresponding to all of WordNet's possibilities as an independent variable of D dimensions, where D corresponds to the number of categories available in WordNet. Then we represented each answer as a value taken by this independent variable, i.e., as a tree labeled with WordNet's categories whose nodes contain the number of occurences of each category in the answer. In our implementation, we used only the nouns present in the answers to build the tree. For example, if we consider the answer "The fur of the tiger is yellow with black stripes", the corresponding value taken by the independent variable would be the following tree:

```
entity: 5
  abstract entity: 3
    abstraction: 3
      attribute: 2
        property: 2
          visual property: 2
            color: 2
              achromatic color: 1
                black: 1
              chromatic color: 1
                yellow: 1
      psychological feature: 1
        cognition: 1
          process: 1
            basic cognitive process: 1
              representational process: 1
                symbol: 1
                  emblem: 1
                    badge: 1
                      chevron: 1
  physical entity: 2
    object: 1
      living thing: 1
        organism: 1
          person: 1
            tiger: 1
    substance: 1
      material: 1
        animal material: 1
          animal product: 1
            animal skin: 1
              fur: 1
```

We used all the answers of a given cluster to determine the mean and the variance of each component of our independent variable. Our goal was to determine which was the best component to describe the meaning of our cluster of answers, so we assigned a score to each component, based on how many times it was activated by words in a given answer cluster. The label given to the cluster would be the label of the component of our independent variable which has the best score. We tried two different score functions:

- $mean * e^{-\sqrt{variance}} * depthofthenodeinthetree$
  This score function gives good results for specific clusters. With our clustering method, the clusters were very general, and hence this score function was not appropriate.
- $mean^2/variance$
  It was the score function which best fit our clustering method, as we were looking for fairly abstract, generic class labels.

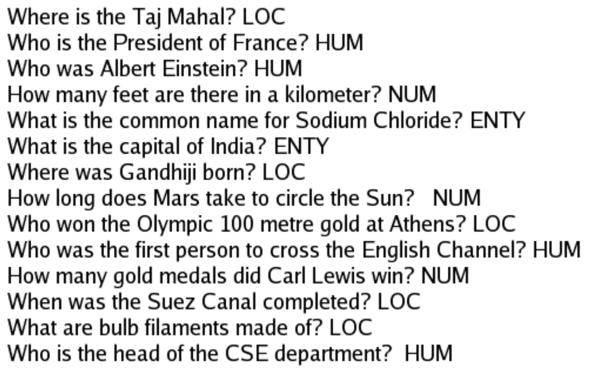
Finally, this method gave the same kind of results as the named entity method; it was able to provide meaningful labels for the coherent clusters that we got, but wasn't too helpful for the others. The main reason was that our clusters were too general to be able to get more precise labels, and to see if this approach was really efficient. However, indications are that with a better set of clusters, the method would be quite useful.

5 Training and Testing

Having gotten cluster labels, using information from both approaches described in the previous section, we were ready to train a classifier. The cluster labels were added to the feature sets for each question obtained earlier, thus forming a labeled dataset which could be employed for supervised learning. This set was then used to train a SVM classifier, with a Radial Basis Function (RBF) kernel. This was also done using Weka; the RBFNetwork classifier therein implements the desired kind of SVM. Once the SVM had been trained, we wished to test it. Ideally, we would have liked a test set consisting only of questions that fell into the kinds of clusters we had managed to obtain, with manually assigned labels. However, making such a set, of size reasonable enough to obtain statistically significant results, would have consumed too much time. So, to be able to get at least a rough indication of how well our classifier was working, we decided to try and use the manually labeled dataset employed for training purposes by Roth and Li [7] (the data is available online at `http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/`). This meant that we had to match our cluster labels with the categories employed by them. We were able to do this for some of the categories, albeit a bit vaguely, and we used only questions belonging to these for testing. Features were extracted for the questions, as described in Section 3, and the classifier was run on them. Its output was compared to the true label (the manually assigned one), and if the two were the same for a given instance, it would be considered a hit.

On a test set of about 1000 questions of the kind described above, our classifier achieved an accuracy of 32%. This is a low figure, and it shows that the method did not work very well. However, as mentioned earlier, the clusters obtained by us did not necessarily have exactly the same labels as those defined by Roth and Li, and in some cases the alignment was done a little arbitrarily. For instance, we obtained 2 clusters with a large number of entities of type PERSON in the answer set. Based on other the other semantic information obtained via WordNet, one of these was given the HUM (human) label from amongst the Roth and Li categories, while the other was given the DESC (description) label. The HUM label was pretty accurate, but the DESC label just reflected the fact that several of the questions in that cluster asked for descriptions of people, though it had other kinds of person-related questions as well. Another factor was that we looked at P_1 accuracy, i.e., only one label was assigned by the classifier to each test question. The inherent ambiguity of many questions makes it difficult to get very accurate results this way; a standard approach is to allow multiple labels to be assigned to each instance, and consider it a hit if, say, the true label is within the first 5 assigned ones (this is known as P_5 accuracy). Since we only had 5 clusters in total, this approach could not be used; it is obviously meaningful only for finer classification.

We did manually construct a small set of 14 questions, and ran the classifier on that see what kinds of labels would be assigned. The results can be seen in Figure 1. Around 10 of the labels are reasonably appropriate. This indicates that better classification accuracy might be achievable on simpler questions, which are closer to the kinds of clusters actually obtained from the training data.



Where is the Taj Mahal? LOC
Who is the President of France? HUM
Who was Albert Einstein? HUM
How many feet are there in a kilometer? NUM
What is the common name for Sodium Chloride? ENTY
What is the capital of India? ENTY
Where was Gandhiji born? LOC
How long does Mars take to circle the Sun?   NUM
Who won the Olympic 100 metre gold at Athens? LOC
Who was the first person to cross the English Channel? HUM
How many gold medals did Carl Lewis win? NUM
When was the Suez Canal completed? LOC
What are bulb filaments made of? LOC
Who is the head of the CSE department? HUM

Figure 1: Classification results on a manually constructed set of simple test questions. The labels mean the following: ENTY is entity, HUM is human, LOC is location and NUM is number.

6 Conclusions and Future Work

The main problem with our method is the difficulty of obtaining quality clusters. Unsupervised learning requires features whose values can be compared for closeness, but this is not possible with the kinds of features used here. Indeed, it appears difficult to obtain semantic features of this kind; one can attempt to use WordNet, as we did, but the distribution of words necessarily will be quite discrete and chunky. Some of the other possible ideas which may be tried out include using named entities as features, and composing more complex features, such as conjunctive (n-grams) and relational ones, from the primitive ones (as done in [7]). Unless a feature set sufficiently rich to allow good clustering can be obtained, our overall approach will clearly not be very useful.

The key novel idea employed here was the use of answers to obtain question labels, which could then be used to train a classifier. As mentioned earlier, this approach is not practicable for single answers, but for large, coherent sets, it seems to have some merit. The combination of information from named entities and WordNet-based semantic analysis allowed us to get fairly meaningful labels for some clusters; the failure to do so in other cases was because there was no single dominant quality to the cluster, i.e. the cluster itself was bad. Thus, clustering is the key bottleneck here; if we get good clusters, then answer analysis can provide good labels. Perhaps a semi-supervised learning approach, where some training instances were manually classified and others were machine-labeled based on the answers, could lead to better results. The problem of natural language question-answering is not an easy one; it has been described by some as AI-complete. We believe that our work adds one more piece to the set; whether it fits into the puzzle or not, only time will tell.

References

[1] M. Collins and M. P. Marcus. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637, 2003.

[2] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods.* The Cambridge University Press, Cambridge, UK, 2000.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[4] C. Fellbaum and C. Fellbaum. *WordNet: an electronic lexical database.* MIT Press, Cambridge, MA, 1998.

[5] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 2002.

[6] S. Harabaigu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. FALCON: Boosting knowledge for answer engines. In *Text Retrieval Conference (TReC)*, 2000.

[7] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the International Conference on Computational Linguistics*, 2002.

[8] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1996.

[9] Ian H. Witten and Eibe Frank. *Data Mining.* Morgan Kaufmann, San Francisco, CA, 2000.