COMP4046 Statistical Natural Language Processing Assignment 2

Enoch Lau
SID 200415765

15 June 2007

Abstract

Question classification may be used as a stage in a system that can answer free form factual questions. The role of such a system would be to help constrain the type of answer that may be produced by a question answering system.

We consider a rule-based question classifier, and then we consider a machine-learning approach to question classification. Both classifiers attempt to classify questions into ”coarse” and ”fine” grained categories. The machine-learned classifier takes into account the syntactic structure of the question, and performs slightly better than the rule-based classifier.

1 Introduction

Question answering (QA) is a task in natural language processing that aims to produce an accurate and concise answer to question. The current focus is on answering questions with factoid answers; for example, we might consider a question such as ”What county is Modesto, California in?” (from TREC 10 [1]). The challenge and the difference between question answering and question classification is that the target text is less likely to overlap with the text in the question [3].

In this report, we discuss the question classification problem, and the chosen classification hierarchy. We then discuss the construction of the rule-based and maximum entropy classifiers, and make comparisons between the approaches taken in both. Finally, we present an evaluation methodology, and evaluate the two classifiers accordingly.

2 Question Classification

With question classification (QC), the goal is to classify a question into a number of predefined semantic classes. This is not a goal in itself; the reason for doing QC is to help restrict the answer type in a question answering problem, thus narrowing the possibilities that need to be considered. As suggested in [3], the more specific the semantic class that we are able to assign, the more beneficial it is for the question answering system; the authors point out that in the past, systems have only been able to support a small number of classes. By considering a syntactic and semantic analysis of the questions, the paper suggests that we may be able to support a larger number of categories.

However, is the support of a more precise classification necessarily beneficial? A problem is that systems in natural language processing are never 100% accurate, and the use of an additional layer before the QA system will introduce additional errors. In particular, one could imagine that an over-precise classification hierarchy would adversely restrict the search space of the QA system, and a more general classification would have allowed the QA system to have chosen the correct answer. A final problem is that the classification of problems is necessarily subjective; because QC is not an aim in itself, the most intuitive classification (for humans) may not necessarily be the correct classification scheme to use to help with QA.

[1] `http://trec.nist.gov/`

3 Classification Hierarchy

We use the classification hierarchy as presented in [3]. This consists of 6 coarse-grained classes, inside of which we find a total of 50 fine-grained classes. The reasons cited by the authors of that paper for including two levels of classification is for compatibility with earlier work (because earlier work only considered a small number of classes) and for a performance advantage over a flat classifier (which was not verified by their experiments). Because of what they termed the ambiguity problem, namely that there is no clear boundary between classes, the authors allowed their classifier to choose multiple classes.

There are various problems with this particular question classification. The classes chosen, in particular the fine classes, seem particularly arbitrary. For example, within the "location" coarse class, there is a "mountain" class but no "ocean" or "park" classes, for example. (On the other hand, a biased set of classes may be valuable when designing a QC system for a QA system that will answer questions in a restricted domain.) This limits the applicability of this question classification schema to the particular set of questions being considered by the authors. Furthermore, the existence of a hierarchy in this instance does not appear to offer much. The fine classes are so disparate that knowing the coarse class a priori would not in general assist in finding the fine category; the authors discovered this experimentally in fact.

4 Rule-Based Classifier

We begin with the hand-created rule-based classifier. The rule-based classifier firstly classifies the questions into coarse classes, and then based on the assigned coarse class, attempts to assign a fine class. To assign the coarse class, we start off by considering the tokens in the question, and assigning scores to each of the coarse classes, based on the words present in the question. Different words are worth different amounts, reflecting their presumed importance in defining the questions that appear in that class. The words and the weights were both chosen by manual visual inspection of the questions in each class, and by the collation of simple statistics, such as the most frequently-occurring words in each class. We originally started with a decision-tree like structure; however, the motivation for using a counting scheme at the top level is that the same decisions about whether a question falls in a particular class ended up being repeated in various points around the tree. There did not appear to be a straightforward structure where the classes could be separated by considering a series of binary questions. We also included words from gazettes to use in the counts; gazettes were available for animals, cities, countries, numbers and occupations, chosen because they posed the most difficulty in generalising to a small number of words that can be manually specified.

After the first stage, errors in the classification were rectified by examining the questions that were misclassified, and seeing if there were any common characteristics between questions of the same type that were misclassified at the same node. Rules were then added in an ad hoc basis. Additional decision criteria were also used in addition to the binary existence of words, such as whether there exists words that consist exclusively of upper-case characters (particularly useful for distinguishing abbreviations), and whether there exist words in the question that start with a particular substring (useful for ignoring variations without the use of a stemmer).

In this classifier, we enforced the rule that the fine class was to be a subclass of the coarse class. A problem with this is that classification on the fine level will suffer because of the already crude classification done at the coarse level. The advantage of enforcing this is that it becomes easier to examine the differences between the various fine classes with a smaller number of questions. Even so, it was difficult and time consuming to come up with a comprehensive set of rules for all 50 classes, and so only the most striking characteristic (mostly the existence of one of a certain set of words) was chosen to classify documents into the fine classes.

5 Maximum Entropy Classifier

Next, we considered a machine-learned maximum entropy classifier. Although the rules from the previous classifier can be used as features, we note that we lose the implicit hierarchy in the rules; however, the decision tree in the rule-based classifier is not particularly deep anyway. In any case, as noted in [3], it appears to be important to consider syntactical information, especially when considering the fine classes.

We used a number of tools in this section. The maximum entropy classifier was MegaM [2], used with default settings. The syntactic information was provided by the C&C tools [1], from which we extracted dependencies, lemmatised forms of words, POS tags, chunks and named entities. The feature sets we used are outlined in the subsections below; different combinations of feature sets can be enabled at a particular time. Note that all features are binary.

5.1 Bag of Words (BOW)

This is the standard bag of words approach, where we take any alphanumeric token and use its presence as a feature. Although the rule-based classifier exists as a baseline as well, this is the most basic approach that can be taken.

5.2 Bag of Words with Stop List (BOW-ST)

Usually, a stop list of common words is used to remove words that are normally considered uninteresting in that they provide little information. For example, ″the″ and ″is″ would be removed. For the purposes of this task, the ″wh″ question words were whitelisted, because intuitively, these words do in fact carry much information about questions even if they are common words. This feature set can be used in place of BOW.

5.3 Gazettes (GAZ)

A number of gazettes were used; these are the same as the rule-based gazettes. The feature is whether or not a question contains a word from a particular gazette. The justification for its use is that the exactly the same words might not be used in a question in the same class, but a word semantically related may be used in its place.

5.4 Bigrams (BI)

English text often contains pairs of words that occur particularly common together, and have a particular meaning attached. In order to capture this, we considered bigrams, and we only took the most significant bigrams, as determined by a t-test with a 99.5% confidence:

$$\frac{p(w_1, w_2) - p(w_1)p(w_2)}{\sigma/\sqrt{n}} > 2.576$$

The standard deviation, σ, was approximated by $\sigma = p$, because we have a binomial distribution and $\sigma = p(1-p) \approx p$ for small p.

5.5 Bag of Words with POS Tags (POS)

POS tags can be used a quick form of word-sense disambiguation, where the same set of word may have different meanings (often used in different parts of speech), and thus this feature set, which joins the words with its POS tag, can be used in place of BOW.

5.6 Named Entities (NE)

Named entities are useful for certain types of questions, such as those involving place names. The named entity recogniser will allow us to pick up on the existence of these types of entities in the question; the bigrams feature set may not include all named entities that are bigrams because they may be rather infrequent.

5.7 ″Wh″ Word Dependencies (WH-WORD)

The idea here is that we capture the dependencies of what the question word is asking about; for example, ″which car″ and ″what year″. The dependency is likely to be the core focus of our response to the question.

5.8 ”Wh” Word Dependencies with POS Tags (WH-POS)

The idea is similar to WH-WORD, only that we consider the POS tag instead of the actual word. For example, it may be the case that a ”which” that depends on a NP is more prevalent in certain classes.

5.9 Gazetted Words with POS Tags (GAZ-POS)

The idea here is that certain classes of words can perform certain actions; for example, occupations may be often used in a particular context in certain classes of questions.

5.10 Long-Range Dependencies (LR-DEP)

The parser allows us to capture long-range dependencies, which we have defined to mean dependent words 4 or more tokens apart. The idea is to capture relationships that are affected by, say, intervening adjectives, such that the bigram feature set will not capture this information.

5.11 Fine-Class Statistics (FINE)

We calculated the most probable words for fine classes, weighted by how uneven their distribution is (measured by way of looking at the variance of their relative frequencies in the classes). The idea is that we need to give more information about fine classes, and what better way than to reinforce words that are more likely to appear in that class than others.

6 Comparison of Rule-Based and Maximum Entropy Classifiers

Once both classifiers have been built, the advantages of a maximum entropy classifier on the development of a question classification system became immediately obvious. Not only was the time to build the maximum entropy classifier shorter than the time to build the rule-based system, the learned classifier was able to absorb new features without extensive manual reconfiguration of the other feature sets. Although building a rule-based classifier may be feasible for the small number of coarse classes, it quickly became a laborious exercise to create rules for the fine grained classes, of which there were 50.

7 Evaluation

The rule-based and maximum entropy classifiers share the same evaluation module. Although the evaluation module calculates macroaverage (equal weight between classes) as well as microaverage (equal weight between questions) scores, we will only consider the microaverage scores here. The reason for this is that the distribution of questions between the classes is not even, and a macroaverage places disproportionate weight on the very small classes, which on average tend to perform poorly due to a lack of data. In addition, because the intention is to build a system that has optimal performance on unseen questions, the microaverage score better reflects this performance, assuming that the distribution of unseen questions matches the distribution of questions that we have been given.

For the rule-based system, the development was carried out by manual examination (as outlined before) on the 5,500 annotated questions from the accompanying website to [3]. For the maximum entropy classifier, we set aside 500 questions as the development test set, and trained the machine learner with the other 5,000; we performed experiments to optimise performance on the development test set. The final test data, from TREC 10, was unseen until the end, and the systems were evaluated against it once only.

Standard measures of performance are precision, recall and F-score, which, as the geometric mean of the two previous measures, balances between them. The microaveraged scores involve taking the confusion matrices of each class and summing them to derive the total count over all classes of true positives, false negatives and false positives. Because a misclassification is a false negative for one class and a false positive for another class, precision, recall and F-score are equal in this context, and thus we present F-score only.

Unlike [3], the classifiers only produced one class per question, instead of taking the top 5. As a result, the results are not directly comparable with [3]. However, we note that their metric is too lenient for the coarse class case, where the classifier reports 5 out of the 6 available classes. In addition, our measurements demonstrate the worst-possible case, should a QA system decide that it is only feasible to consider one class.

8 Results

8.1 Rule-Based Classifier

On the 5,500 training set, we achieved a microaveraged F-score of 0.579 for coarse classes, and 0.466 for fine classes. On the TREC 10 test set, we achieved a microaveraged F-score of 0.694 for coarse classes, and 0.568 for fine classes.

8.2 Maximum Entropy Classifier

We examined a range of combinations of feature sets, and the results for the microaveraged F-score is presented in the table. The combinations of feature sets are as described in the corresponding subsection above.

	Coarse	Fine
BOW	0.690	0.564
BOW-ST	0.666	0.548
BOW + GAZ	0.812	0.558
BOW + GAZ + BI	0.812	0.558
BOW + GAZ + POS	0.736	0.572
BOW + GAZ + POS + NE	0.760	0.580
BOW + GAZ + POS + NE + WH-WORD	0.858	0.602
BOW + GAZ + POS + NE + WH-POS	0.770	0.598
BOW + GAZ + POS + NE + WH-WORD + GAZ-POS	0.814	0.600
BOW + GAZ + POS + NE + WH-WORD + LR-DEP	0.750	0.618
BOW + GAZ + POS + NE + WH-WORD + FINE	0.620	0.496

The final test on the TREC 10 set was performed with the options BOW + GAZ + POS + NE + WH-WORD + LR-DEP, which scored the highest score for the fine class task. On the TREC 10 data, we achieved 0.802 for coarse classes and 0.644 for fine classes.

9 Future Work

Based on the work performed, we suggest further avenues of exploration:

- Different combinations of features: we cannot assume that the feature sets used for the maximum entropy model are independent of each other, and we have only considered a small number of combinations in the experiments above.
- Separating the features between the coarse and fine grained classifiers: as we can see from the table, the coarse class F-score peaked at 0.858, but improving the fine-grained analysis with more syntactics-based feature sets decreased the performance for coarse classes. If we also use the coarse class information in the fine class stage, we may be able to improve the performance for coarse and fine classes independently.

- Perform experiments using the P_5 measure in [3]: this will allow us to compare our results with the results in that paper.

10 Conclusion

We considered a rule-based and a machine-learned maximum entropy classifier, using the classification hierarchy from [3]. The maximum entropy classifier performed better than the rule-based classifier, and we achieved a final microaveraged F-score of 0.802 for coarse classification and 0.644 for fine classification.

References

[1] S. Clark and J. R. Curran. Parsing the wsj using ccg and log-linear models. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 103, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[2] H. Daumé III. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name`, implementation available at `http://hal3.name/megam/`, August 2004.

[3] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.