CS229 Final Project: Enhancing Automated Question Classification

William Mee wmee@stanford.edu
Seung-Yeoul Yang syyang@stanford.edu

10 December 2010

Abstract

This class project investigated improving classification of short questions. We first established a baseline for further work by comparing a number of different off-the-shelf classifiers. We gathered a new corpus to add to ones already available. We then implemented and applied semi-supervised classification, adding large amounts of unlabelled data in an attempt to boost classification accuracy. Finally we investigated feature expansion, both via semantic knowledge and with augmentation by automatically learned topics.

1 Introduction

This work investigated classification of short, closed questions. Examples of these are

> What was the United States' first national monument?
> How do you say red in Spanish?

Apart from being short, such 'factoid' questions are well-structured: they generally have a question word, one verb and one or two nouns. This makes question classification a domain distinct from document classification. The question classification task is important as a part of a larger automated question and answer system.

We began by analyzing the effectiveness of several classifiers, including Naive Bayes and Support Vector Machines on labelled training and test questions. We then used this as a baseline to explore the following series of experiments to improve the accuracy of the classification

1. A semi-supervised approach in which we applied Expectation Maximization (EM) on a combination of labelled and unlabelled questions
2. Feature expansion using lexical information
3. Feature expansion using automatically learned topics

Details of these experiments follow.

2 Question Datasets

We made use of two corpora for this project. Firstly, tests were done on the TREC-10 QA dataset [7]. This data has 5500 short questions which are labelled with 6 coarse categories (Abbreviation, Description, Entity, Location, Human and Number) and 50 fine subcategories. An example question from this dataset is

> When did the Berlin Wall go up ?

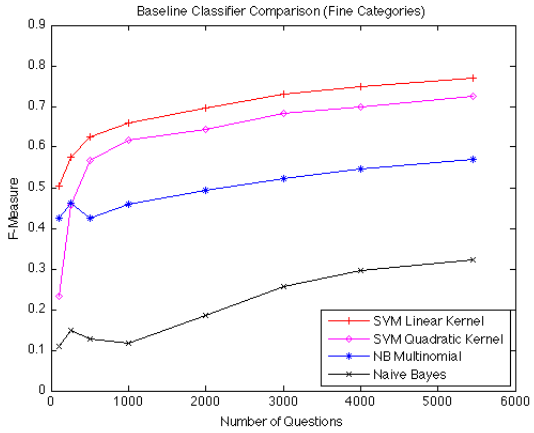
which is assigned to the coarse:fine category **NUM:date**.

The second corpus is one which was gathered for this work from Yahoo! Answers [9], the only online question system we are aware of which has a public API. Answers data is structured into a hierarchy of themes, for example "Environment/Global Warming". We selected questions from the History theme, since many of these are in closed form, and manually labelled the data using the same categories as the TREC-10 questions. The english questions downloaded were those marked as 'resolved', reverse sorted by number of responses. Although we filtered out questions which were not in closed form and de-duplicated, we did not do spelling correction or make grammatical changes; as such this Yahoo! History data is significantly less regular than the TREC-10 dataset.

3 Comparison of Supervised Classifiers

An important initial step was to establish a baseline comparison for different supervised classifiers on short questions. For algorithm implementations, we made use of the Weka Machine Learning Project[8] . This allowed us to quickly compare three different classifiers (Naive Bayes, Naive Bayes Multinomial and SVM). It also allowed us to experiment with the effects of feature selection, stopwords and stemming. In initial experiments, we ran the candidate classifiers against training subsets of various sizes and measured the error rate of the resultant classifier by using 10-fold cross-validation against the same training data.

Figure 1: Baseline Classifier Comparison, Fine Categories



3.1 Feature Selection, Stemming and Stop Words

We experimented with using Mutual Information measures to reduce the numbers of features. In all experiments, this increased the error rate of the categorizing, and the fewer features used the more the error rate increased. For example, selecting the top 300 features for the Naive Bayes Multinomial on a training set of 5500 questions reduced accuracy from 76.4% to 75.6%; using only 100 features further reduced this to 69.5%. One explanation of this is that the questions being categorized are very short, so almost all the limited number of features have some discriminating use. We used all features for the rest of this project.

We also experimented with using a set of stop words obtained from the Apache Lucene project. Similarly to the feature reduction, in all experiments leaving out the stop words slightly degraded the performance of the classifiers. For example, the accuracy of Naive Bayes Multinomial on 5500 questions decreased from 76.4% to 75.4%

While some experiments saw a decrease in accuracy when using stemming, most had an improvement. We therefore consistently used stemming for the rest of this project.

3.2 Comparison Results

The different classifiers displayed different performance characteristics with training set size. A summary of our baseline is given in Figure 1; the data had stemming applied to the fine categories in the TREC-10 dataset. It is unsuprising that that the SVM with linear kernel provides the best accuracy for any input dataset size, and that accuracy increases with dataset size. However since one of the core experiments we performed (semi-supervised learning using Expectation Maximization) is based on a Naive Bayes Multinomial classifier, we used this classifier for comparison purposes in many other experiments.

4 Semi-Supervised Classification Using EM

Semi-supervised classification combines labelled and unlabelled data to improve classification accuracy. While this is an active field of research in machine learning in general [10], there is little application of this to the question-answer domain. However, the intuition that additional structure implicitly provided by unlabelled data which could help classification seemed applicable. A key motivation is that with question classification, or text classification in general, there is an abundance of unlabelled input data, which we would like to take advantage of for improving our classifier. Our own data collection effort was proof of how difficult and tedious data labelling can be.

In this part of the project, we implemented the combination of a Naive Bayes Multinomial classifier with Expectation Maximization in two variations suggested by Nigam et. al[6]. The basic approach here is a generative one, which supposes every document is generated from a probabilty distribution parameterized by θ. The probabilty distribution consists of a mixture of components $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$. In addition, the training data $\mathcal{D}$ consists of labelled data, $\mathcal{D}^l$, and unlabelled data, $\mathcal{D}^u$, such that $\mathcal{D} = \mathcal{D}^l \cup \mathcal{D}^u$. Parameter estimation is done with the labelled data and then repeatedly refined by applying to the unlabelled data and retraining in a form of Expectation Maximization:

---

1. Build an initial classifier from $\mathcal{D}^l$ only and calculate parameters $\hat{\theta}$
2. Repeat until convergence:
   E-Step: Use the current $\hat{\theta}$ to classify $\mathcal{D}^u$
   M-Step: Re-estimate $\hat{\theta}$ given estimated component membership of all data

---

This approach is mathematically correct only for a Naive Bayes Multinomial classifier. This base algorithm is then refined in two variants detailed below.

4.1 EM-λ

Assuming there is one-to-one correspondence between a class and a mixture, i.e., each class is generated by a single mixture, the EM-λ algorithm suggested by Nigam et al allows the contribution of unlabelled data to be weighted to avoid dominance when the ratio of labelled to unlabelled data is low. In this scenario the log likelihood equation can

be stated as:

$$\ell(\theta|\mathcal{D};z) = \log P(\theta) + \sum_{d_i \in \mathcal{D}^l} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log(P(c_j|\theta)P(d_i|c_j;\theta)) + \lambda \left(\sum_{d_i \in \mathcal{D}^u} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log(P(c_j|\theta)P(d_i|c_j;\theta)) \right)$$

where z_{ij} the probability of d_i belonging to c_j; and λ, $0 \leq \lambda \leq 1$, determines the contribution of the unlabelled data to the overall log likelihood.

4.2 Multiple Mixture Component per Class

The multiple mixture component per class approach relaxes the assumption in EM-λ, allowing for a many-to-one correspondence between mixture components and classes. This is motivated by an effort to model documents as close to reality as possible, since the number of latent variables may be greater than the number of classes. To find the posterior probability $P(c_j|d_i)$, once we obtain the probability for each mixture component, we then need to sum the probabilities over the classes to which each mixture component belongs. Note in our implementation, the number of mixture components is set to be identical for each class. Furthermore, we heuristically pick a small number for the number of mixture components since having a large number of mixture components drastically increases the training time.

4.3 Test Results

We tested both the EM-λ and multiple mixture model against UIUC dataset and questions collected from Yahoo! History data. However, neither of them managed to achieve a significant improvement over off-the-shelf classifiers. In fact, our experiments showed that in the context of question classification, incorporating unlabelled data can actually deteriorate classification performance.

As can be seen in Figure 2, the result of EM-λ was worse than the baseline performance of Naive Bayes Multinomial. The result was obtained in 10-fold cross validation.

Figure 3 compares the performance of baseline Naive Bayes Multinomial classifier with multiple mixture component classifiers of varying mixture component size, denoted as k. Unfortunately, the algorithm did not do too well on our data sets. We tried running the classifier with a higher k such as 6, 8 and so on, but all of them gave the same result as in the case when $k = 2$. One possible explanation for the poor performance may be that questions violate the model assumption enforced by multiple mixture component model.

Figure 2: EM-λ Test Results

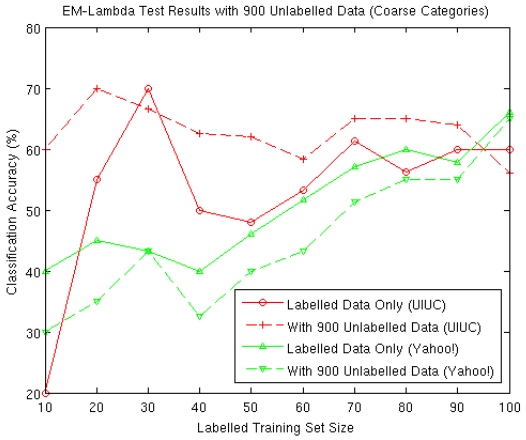
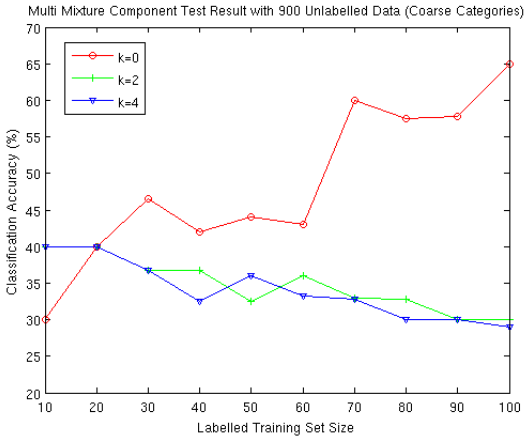


Figure 3: Multiple Mixture Component Test Results



5 Semantic Expansion

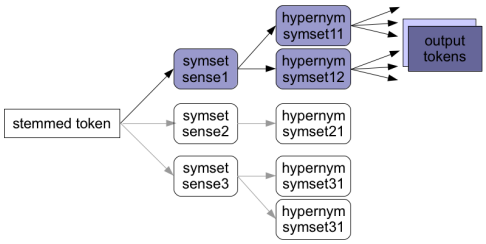
In this section of our project, we investigated augmenting features of the dataset with ones obtained from semantic structure. The intuition here is that, because the factoid questions are almost all single, short sentences, the probability of a feature being present in both training and test questions selected from same category is much lower than in longer documents, and augmenting features with more generalized version would improve classification accuracy.

Our approach was to to identify nouns and verbs in the questions, and then expand these with hypernyms which provide generalizations. For example, the hypernyms of 'car' include both 'motor vehicle' and 'compartment'. Following similar experiments [3] [2] we used a part-of-speech tagger in an initial step, and then used a sequence of WordNet[4] lookups to do the expansion. WordNet maps a word to one or more senses, each of which is associated

with a 'symset' of words which share the same set; each symset is in turn linked via hypernym (and other) pointers to other symsets.

The tokens we chose to expand, as well as the expansion policy within WordNet, were investigated in a series of experiments. In terms of selecting candidate token, we looked at generalizing all nouns, all verbs and then just the first noun as a substitute for the 'lead' noun, as suggested by [3]. We then experimented with aggressive expansion, in which all senses of the word were generalized, versus conservative expansion, in which only the first sense of the word was generalized (see Figure 4).

Figure 4: Hypernym Expansion, with conservative approach highlighted



5.1 Test Results

Classification accuracy was consistently improved by hypernym expansion of nouns, with the best results obtained in aggressively expanding a single noun only, as shown in Figure 5. The improvement was measured across classifiers.

The impact of expanding verbs in a similar way had an insignificant impact on accuracy; possible explanations include that the hypernym network is less dense in WordNet for verbs than for nouns and that the verbs are less useful for descrimintation; indeed in a large number of the questions, the only verb is 'to be'.

The positive impact that the hypernym expansion has on frequency of token occurrence is demonstrated in Figure 6.

The heuristic of using the first noun as the lead noun is possible with the well-structured question dataset, but would not be applicable in a normal text with longer sentences.

In a second set of experiments, we applied the same expansions to the Yahoo! History 1000-question dataset. The baseline classification accuracy using Naive Bayes Multinomial on this data was 43.5%, which is comparable to the 44% accuracy on the TREC-10 data although the number of categories dropped to 37 from 50. Details of the improvement given by hypernym expansion is given in Table 1.

Figure 5: Single-Noun Hypernym Expansion Test Results

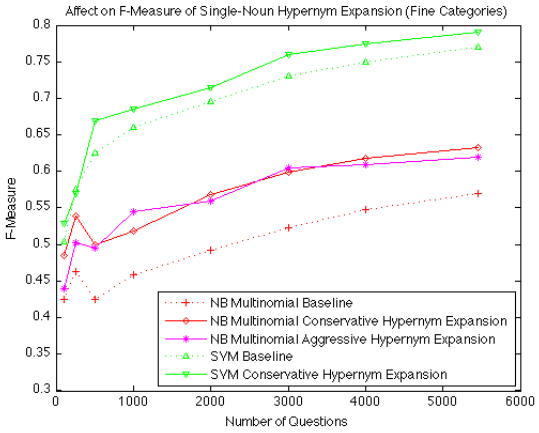
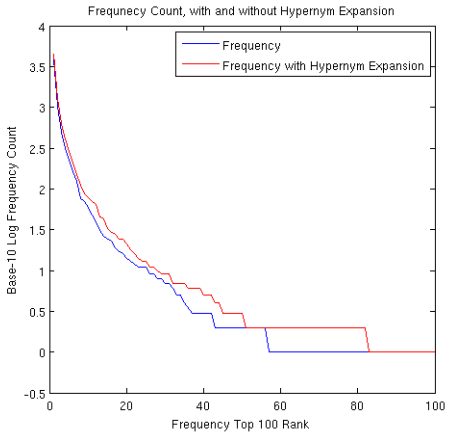


Figure 6: Token Frequencies with Hypernym Expansion of Nouns



6 Latent Dirichlet Allocation

Another classification method we considered was latent Dirichlet allocation (LDA). LDA represents a document using a generative model where given a class, the document is generated from multiple mixture components . Note that LDA relaxes the many-to-one correspondence assumption in the multiple mixture component model, discussed in the previous section, by assuming a many-to-many correspondence between mixtures components and classes. Experiments have shown that the weakened assumption improves classification for text documents [5].

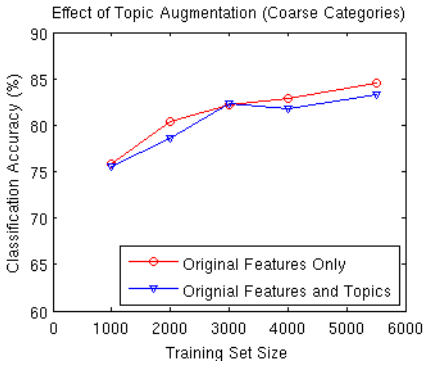
For our purpose, we used topics obtained from LDA to augment the features in each training example to improve classification performance. Using the Mallet software package, we first ran LDA on the training examples to extract topics from the training examples (the number of the topics was empirically set to 300) [1]. We then augmented the

Table 1: Hypernym Expansion of Yahoo! History Data

Description	Accuracy	Precision	Recall
Baseline NB Multi.	43.5	0.38	0.44
Conservative Hyp. Exp.	45.5	0.41	0.46

original features with the topics, and trained an SVM with polynomial kernel on the new data. However, as shown in Figure 7, adding the topics to the original features actually degraded the performance slightly. In hindsight,this is somewhat expected because we were not adding any new information to the features.

Figure 7: LDA Topic Augmentation Test Results



7 Discussion and Conclusion

In this project we broadly examined approaches to improving classification of closed-form 'factoid' questions. An initial examination of the impact of stemming, stop words and feature reduction and comparison of several well-known supervised classification techniques allowed us to establish a baseline to measure later work on.

One of the core aims, the application of semi-supervised classification by adding a large amount of available non-labelled data to a limited set of labelled data, was examined in the form of Expectation Maximization which has successfully been applied to datasets of documents. We both implemented multiple versions of this technique, but failed to use it successfully, presumably because the generative model on which this approach is based, while it may still apply to questions, is not measurably manifested because of question length.

This lead to investigations of how the limited features available could be augmented. We did this both through application of external semantic knowledge and with topics automatically learned from the dataset using latent dirichlet analysis. While the latter did not prove effective, noun hypernym expansion based on a simple heuristics was shown to consistently improve classification accuracy across classifiers and granularity of categories. Use of more sophisticated word-sense disambiguation and better identification of the lead noun of the sentence is likely to improve this more, but was beyond the scope of this project.

Our work demonstrated how some aspects of machine learning are robust across different datasets, while others are restricted, sometimes in subtle ways, which limits their applicability. Both the failure of semi-supervised classification and the effectiveness of semantic feature augmentation were unanticipated results of this work.

References

[1] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. URL http://mallet.cs.umass.edu, 2002.

[2] Xin Li and Dan Roth. Learning question classifiers: The role of semantic information. In *In Proc. International Conference on Computational Linguistics (COLING*, pages 556–562, 2004.

[3] Donald Metzler and W. Bruce Croft. Analysis of statistical question classification for fact-based questions. *Journal of Information Retrieval*, 8:481–504, 2004.

[4] George Miller. Wordnet A lexical database for English. *Communications of ACM*, 38(11):39–41, 1995.

[5] David M. Blei Andrew Y. Ng and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. In *Machine Learning*, pages 103–134, 1999.

[7] NIST TREC-10 QA Track. `http://trec.nist.gov/data/qa/`.

[8] Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/~ml/weka.

[9] Yahoo! Answers. `http://answers.yahoo.com/`.

[10] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.