interspeech 2008 incorporating SST 08

Inductive and Example-Based Learning for Text Classification

Ye-Yi Wang, Xiao Li and Alex Acero

Speech Technology Group, Microsoft Research, Redmond, WA, USA

{yeyiwang,xiaol,alexac}@microsoft.com

Abstract

Text classification has been widely applied to many practical tasks. Inductive models trained from labeled data are the most commonly used technique. The basic assumption underlying an inductive model is that the training data are drawn from the same distribution as the test data. However, labeling such a training set is often expensive for practical applications. On the other hand, a large amount of labeled data, which have been drawn from a different distribution, is often available in the same application domain. It is thus very desirable to take advantage of these data even though there is a discrepancy between their underlying distribution and that of the test set. This paper compares three text classification algorithms applied in this scenario, including two inductive Maximum Entropy (MaxEnt) models, one flatly initialized and the other initialized with a term-frequency/inverse document frequency (Tf*Idf) weighted vector space model, and an example-based learning algorithm, which assigns a class label to a text by learning from the labels assigned to the training data that are similar to the text. Experiment results show that example-based learning has achieved more than 5% improvement in precisions across almost all coverage levels.

Index Terms: text classification, inductive models, maximum entropy model, Tf*Idf vector space model, example-based learning.

1. Introduction

Text classification has been widely applied to many practical tasks. In spoken dialog systems, it has been used in call-routing [1], sentiment classification [2], dialog-act tagging [3] and answer machine detection [4], etc. Similar to call-routing in speech applications, text classification has also been used to understand users' intent in web search and route their queries to appropriate search verticals (e.g., job search, local search, product search, etc.)

Inductive models like Naïve Bayes classifier, Support Vector Machines (SVMs), Boosting or MaxEnt models are commonly used for text classification [1],3,5]. They are trained from labeled training data, under the assumption that the data are drawn from the same distribution as the test set. Manually labeling such a training set is often expensive for practical applications. It is also difficult when the label set contains more than a few destination classes – inter-labeler agreement rate is often low in this case, which leads to inconsistently labeled training data. On the other hand, a large amount of labeled data, which may have been drawn from a distribution that is discrepant from that of the test set, is often available in the same application domain. For example, data from email or chat support, or even from a related website, can be leveraged to train an IVR system to route an incoming call to an appropriate service department. Review data from the web can be used to train a sentiment classifier for spoken dialog systems like Voice-Rate [2], which allows callers to get or leave reviews for a product or service. Business or product listings from a database are often used to model users' spoken query in voice search applications [2, 5]. In the vertical job search on the web, a user's query is classified into its corresponding category, and only the job listings with the matching category will be displayed to the user. While it is costly and difficult to manually label users' queries with 30+ categories, a large amount of listing data with labeled categories can be automatically harvested from job sites like www.careerbuilder.com. However, the listing names often appear different from real queries, as demonstrated by the exemplar listing "*senior level sales– interior home products-product exp not req*."

A common practice to address training/test distribution discrepancy is via model adaption [3], which needs a small amount of labeled training data drawn from the same distribution as the test data. This paper looks at the problem from a different angle and proposes example-based learning, which makes classification decisions based on the class labels of the training samples similar to a test query. Contrary to the inductive learning, the training data that bear little similarity to a test query do not contribute much to the decisions in this case. For those samples that a decision is based on, their similarity to the test query makes them closer to the test data distribution. Hence it is less subject to the distribution discrepancy problem and does not require additional labeled training data. We compare its performance with two inductive MaxEnt models, one flatly initialized (all parameters are set to 0 initially) and the other initialized with a Tf*Idf weighted vector space model. Experiment results have shown that the example-based algorithm consistently outperforms both inductive models by achieving better precisions across different coverage levels.

The paper is organized as follows. Section 2 reviews the three different statistical classification algorithms that we investigated in this paper. Section 3 describes the experiment setup and presents experimental results. Section 4 analyzes the strength of example-based learning and proposes future work. Section 5 concludes the paper.

2. Text Classification

2.1. Inductive Model –Maximum Entropy Classifier

A MaxEnt classifier builds the probability distribution $P(C|Q)$ from a set of features $\mathcal{F}$, where C is a random variable representing the classification destinations, Q is a random variable representing the input text to the classifier. A feature in $\mathcal{F}$ is a function of C and Q. The classifier picks a distribution $P(C|Q)$ to maximize the conditional entropy $H(C|Q)$ from a family of distributions, with the constraint that the expected count of a feature predicted by the conditional distribution equals to the empirical count of the feature observed in the training data:

$$\sum_{C,Q} \hat{P}(Q)\,P(C|Q) f_i(C,Q) = \sum_{C,Q} \hat{P}(C,Q)\, f_i(C,Q), \forall f_i \in \mathcal{F}$$

where $\hat{P}$ stands for empirical distributions in a training set.

The maximum entropy distribution that satisfy the above equation have the following exponential (log-linear) form, and the model parameters can be estimated by maximizing the conditional probability of a training set of C and Q pairs [6]:

$$P(C|Q) = \frac{1}{Z_\lambda(Q)} \exp\left(\sum_{f_i \in \mathcal{F}} \lambda_i f_i(C, Q)\right)$$

here $\lambda_i^{'}s$ are the model parameters, also known as the feature weights, and $Z_\lambda(Q) = \sum_C \exp\left(\sum_{f_i \in \mathcal{F}} \lambda_i f_i(C, Q)\right)$ is a constant that normalizes the conditional distribution. The model parameters can be estimated with an iterative procedure that starts from an initial parameterization and gradually updates it towards the optimum. We used Stochastic Gradient Ascend (SGA) in our experiments [7].

2.2. MaxEnt Model Parameterization with TF*IDF Weighted Vector Space Model

The TF*IDF weighted vector space model [8] measures the similarity between a query and a document as the cosine between the two vectors representing the query and the document. Here each element of the vectors represents the importance of a term (e.g., a word or a bigram). The importance increases proportionally to the number of times a term appears in a query or a document, and decreases when the term appears in more documents (hence becomes less discriminative). The *term frequency* (TF) $\mathrm{tf}_i(q)$ and $\mathrm{tf}_i(d)$ are the relative frequency of term i in q and in d, respectively. The *inverse document frequency* (IDF) is the logarithm of the total number of documents divided by the number of documents that contain i:

$$\mathrm{tf}_i(q) = \frac{n_i(q)}{\sum_k n_k(q)}, \qquad \mathrm{tf}_i(d) = \frac{n_i(d)}{\sum_k n_k(d)},$$

$$\mathrm{idf}_i = \log|D| - \log|\{d: i \in d\}|$$

where $n_i(x)$ is the number of occurrences of term i in x, and D is the entire document collection. The weight for term i in a vector is the product of its TF and IDF score.

The cosine score between q and d can be written with respect to each term i:

$$\cos(q, d) = \frac{q \cdot d}{\|q\|\|d\|} = \sum_{i \in q} \frac{\mathrm{tf}_i(q) \times \mathrm{idf}_i \times \mathrm{tf}_i(d) \times \mathrm{idf}_i}{\|q\|\|d\|}$$

$$= K(q) \sum_{i \in q} \frac{\mathrm{tf}_i(d) \times \mathrm{idf}_i^2}{\|d\|} f_{d,i}(d, q)$$

Here $1/\|q\|$ is absorbed by $K(q)$. The term frequency $\mathrm{tf}_i(q)$ is replaced by the integer feature value $f_{d,i}(d, q)$ (number of occurrences of the term i in q) because they differ by a factor of $|q|$, which is also absorbed by $K(q)$. This equation suggests that the vector space model be viewed as a linear classifier, where each document is a destination class, and the feature $f_{d,i}$ and its weight are

$$f_{d,i}(c, q) = \begin{cases} |\{k: q_k = i\}| & \text{if } c = d \\ 0 & \text{otherwise} \end{cases}, \quad \lambda_{d,i} = \frac{\mathrm{tf}_i(d) \times \mathrm{idf}_i^2}{\|d\|}$$

$K(q)$ is negligible in the weight here because it does not affect the classification boundary. For a more typical classification task where multiple training sentences have the same class label, they can be concatenated to form a super "document" representing the class, and the Tf*Idf weighted vector space model can be used as an inductive linear classifier.

In [9], the weights from this linear classifier have been imported to initialize a MaxEnt model and used as the mean values of the Gaussian distributions that regularize the model's parameters. It has been shown that even though a MaxEnt model converges to the global optimum in theory thanks to the convexity of its objective function, this initialization/parameterization scheme can still significantly improve classification accuracies in practice, especially when only a small set of training data is available. This is due to the fact that convergence is determined empirically, and early stopping of training is a common practice to avoid over-training. The discriminative power of the Tf*Idf weighted vector space model, together with its inherent weight sharing mechanism, enables more robust performance of the MaxEnt model it initialized, especially when training set is small.

2.3. Example-Based Learning

Unlike inductive learning, which produces a model from a set of training data and use the model to label unseen test data drawn from the same distribution, example-based learning is a learning paradigm that labels test data according to how the similar data in the training set are labeled. Here "label" is in a more general sense than classification destinations. It can be a sequence of labels in tasks like part-of-speech tagging or named entity identification; it can also be the counterpart of a sentence in a different language, as in the example-based machine translation [10].

Recently there have been increasing interests in graph-based learning in speech and language applications. Graph-based learning is a special case of example-based learning, which is primarily used for semi-supervised learning. In this scenario, the training set is composed of a small set of labeled data and a large set of unlabeled data. Both labeled and unlabeled samples are represented in a graph. Each node in the graph represents a data sample; different nodes are connected by edges associated with a weight representing the similarity of the samples. Based on this graph, class labels can be assigned to the unlabeled nodes in the graph according to the labels assigned to the connected nodes in the graph and the weight of the connecting edges. The similarity can be the cosine value between two samples, as seen in the vector space model, or the resemblance of the click-through patterns for different queries in a classification task that decides if a query has a specific intent [11]. Graph-based learning leverages the similarity not only between train and test data, but also within the test data itself, so it can produce a model that better "explains" the test data. It also uses the unlabeled test data to find better feature representation for learning. So in [12], graph-based learning is applied for phonetic classification in a transductive framework, where an initial model is adapted with an unlabeled test set, such that the classification accuracy on this test set can be improved.

For those training samples that a classification decision is based upon in example-based learning, we believe that their similarity to a test query makes them less prone to the discrepant distribution problem. Therefore we have applied example-based learning for the job category classification task with the following algorithms, where the training data (job listings s and their category labels $l(s)$) were obtained from three job posting websites:

Input: a query q, a similarity threshold t
foreach sample-label pair $(s, l(s))$ in the training set
  similarity = $\cos(q, s)$
  update $\mathrm{score}(q, l(s))$ according to *similarity* and t
Output: $\mathrm{argmax}_l(q, l)$ as the class label for q.

Three different methods are used to update the score for

assigning a class label l to q:

1. Maximum:
   $\text{score}(q,l) = \max_s \cos(q,s)\,\delta\big(l = l(s)\big)$.
2. Majority Vote:
   $\text{score}(q,l) = \sum_s \delta\big(l = l(s)\big)\delta(\cos(q,s) > t)$.
3. Posterior:
   $$\text{score}(q,l) = \frac{\sum_s \delta\big(l = l(s)\big)\delta(\cos(q,s) > t)\cos(q,s)}{\sum_s \delta(\cos(q,s) > t)\cos(q,s)}$$

Here $\delta(x) = 1$ when $x = \text{true}$, and 0 otherwise. The posterior scoring can also be called "Sum" since the denominator is independent of l, so it would not affect the classification result.

In practice, inverted index can be built to locate listings containing a specific term, such that only the listings containing a term in a test query will be involved in example-based learning. This makes the algorithm fast enough for online services.

3. Experiments

3.1. Experiment Setup

We have applied the three learning algorithms described in the previous section to the task of job category classification. There are 33 job categories (e.g., *Accounting, Healthcare*, *IT/Software, Management*, etc.) in Microsoft Live Search's job taxonomy. 587 job related user queries have been manually labeled with their categories, which were split into test (400) and development (187) set.

188,436 job listings were extracted from 3 different job posting web sites for model training. The job listings are posted under their corresponding categories, so there is no need to manually label the category for these listings. Instead an automatic mapping tool converts the category names used in these websites to the 33 categories in our taxonomy. The automatically obtained category labeling is ambiguous and noisy. For example, "accounting manager" is posted in both the "Accounting" and the "management" categories, or even "IT/Software" if the employer of the job is an IT/Software company.

Among all the 188,436 job listing instances, there are 76,166 unique sample types. 39,744 of them have ambiguous labels, which covers 126,292 sample instances – i.e., about 2/3 of the data are ambiguously labeled. The labels for each sample type forms a statistical distribution over the label set. The average entropy of the distribution is 0.33. This implies that each sample type statistically has 1.26 category labels on average.

It is apparent that the job listings and the actual users' queries have not been drawn from the same distribution. Table 1 shows the first 5 IT/Software jobs in both training (job listing) and test (query) data. It is very unlikely for users to issue a query like the third and the fifth job listings in the table. The odds for the first two listings are higher, but they still sound a bit strange to be a real query. Only the fourth listing, "mysql dba," appears like a real query.

Several experiments have been conducted with the data sets. In the first experiment, we have attempted to automatically select a subset of training samples that looks like real queries, and built MaxEnt models on the subset. Different sample selection heuristics have been tried, ranging from sample lengths to number of special characters (exclamation marks, parenthesis, etc.) in a listing, which result in the selected subsets that contain 1/8~1/2 of the listings in the original training set. Unfortunately, none of MaxEnt models built on these subsets of training data have accomplished better accuracy than the model trained with the entire training set. This reveals that the listing data are still helpful in improving the classification accuracy even though they were drawn from a different distribution – *no data is like more data*.

Table 1. *First 5 IT/Software job listings and queries.*

Job Listings	Queries
senior engineer, python/ajax	testing jobs hyd
javascript developer (part time/contract)	computer operator jobs
3d graphics software engineer (c++) at daz 3d	government it jobs
mysql dba	sas jobs
lead web solutions engineer- gap inc. direct	computer engineering jobs

3.2. Experimental results

The second experiment compares the three different scoring methods for example-based learning, as described in Section 2.3. The similarity threshold t in the algorithm was tuned separately for the three scoring methods with the development set.

Instead of comparing the classification accuracy among the three scoring methods, the precision-coverage curves are compared – in the actual application an operational point needs to be selected for the trade-off between the relevancy of the jobs shown for a query (the precision of classification) and the percentage of job queries that result in the showing of the jobs in a specific category (the coverage of classification.) The precision-coverage measure is closely related to the more common precision-recall curve. Both are obtained by varying the threshold for accepting the automatic classification results. The accuracy of classification, which was studies in [9], is the precision at 100% coverage level.

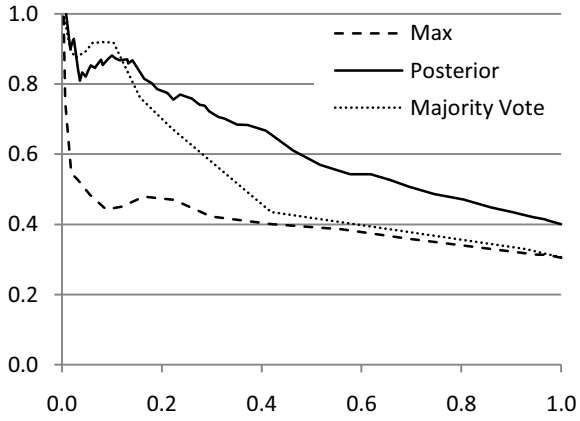

Figure 1: Precision-coverage curves for three different scoring methods in example-based learning. X-axis represents coverage, y-axis represents precision.

Figure 1 shows the precision-coverage curves on the test set for the three scoring methods in example-based learning. Posterior scoring is a clear winner when the coverage is greater than 15%. For maximum and majority vote scoring, although they have the same accuracy, the precision-coverage curves are very different. Maximum scoring outperforms majority vote across all coverage levels, especially in the low coverage region. The best accuracy is at 40%, achieved by the

posterior scoring. The relatively low accuracy is primarily due to the complexity of the task (33 target classes) and ambiguously labeled training set.

The third experiment compares the performance of the inductive learning, including the MaxEnt models with flat initialization and with the Tf*Idf weighted vector space model initialization, and the example-based learning for job category classification. Here the MaxEnt model with flat initialization has the same accuracy as the MaxEnt model initialized with the imported Tf*Idf vector space model parameters. However, the latter has a better precision-coverage curve than the flatly initialized model. The example-based learning achieves the best precisions, which improves over the flatly initialized MaxEnt model by at least 5% absolutely across almost all coverage levels.

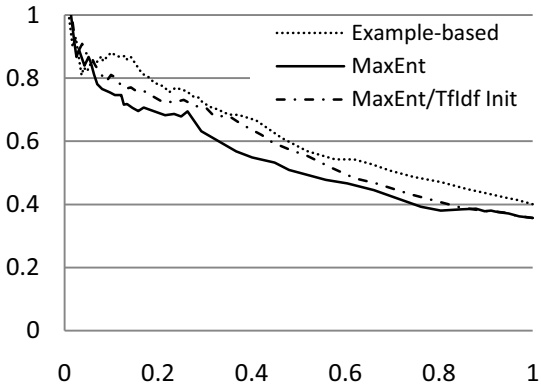

Figure 2. Precision-coverage curves of the MaxEnt classifiers, with flat initialization and Tf*Idf weighted vector space model initialization, and of the example-based learning algorithm. X-axis represents coverage, y-axis represents precision.

4. Discussion and Future Work

As we have expected, the improvement of the example-based learning over the inductive models can be attributed to the fact that it bases its decisions more on the samples that are similar to test data. Hence it is less subject to the problems caused by distribution discrepancy. This is manifested by the experimental result that compares the three different scoring methods in the example-based learning algorithm: the majority vote blindly treats all the training samples equally as long as they are above the similarity score threshold. Hence it is more subject to the distribution discrepancy, which results in poor performance. The posterior scoring, on the other hand, bases its decision more on the training samples with higher similarity score, i.e., samples that follow the similar distribution as the test data. Therefore it has achieved better performance. The maximum scoring, when applied with higher similarity threshold (in the lower coverage region), makes classification decision according to only those highly similar training samples, so it has the best precisions. As the threshold is getting lower (coverage is getting higher), the similarity between the training samples and the test data becomes lower, which implies that they are less likely to be drawn from the same distribution. Therefore its performance is overtaken by the more robust decisions based on more samples with the posterior scoring method.

For future work, we would like to extend the example-based learning to graph-based learning when more unlabeled query data become available. This can be accomplished by applying example-based learning to label the query data, which is less subject to the distribution discrepancy, and then train inductive models from the automatically labeled query data.

5. Conclusions

This paper has investigated the classification algorithms for practical tasks when the training and test data are drawn from different distributions. It compares two inductive learning algorithms, namely the MaxEnt models with flat initialization and the Tf*Idf weighted vector space model initialization, with an example-based learning algorithm. Experiment results have shown that the example-based learning using posterior scoring has achieved better precision-coverage performance over the inductive models. This can be attributed to the fact that example-based learning bases its decision more on the training samples that are similar to the test query. Therefore distribution discrepancy has less significant negative impact on classification performance.

6. References

1. Gilbert, M., et al., *Intelligent Virtual Agents for Contact Center Automation.* IEEE Signal Processing Magazine, 2005. **22**(5): p. 32-41.
2. Zweig, G., et al., *The Voice-Rate Dialog System for Consumer Ratings*. Proceedings of *INTERSPEECH*. 2007: Antwerp, Belgium. p. 2713-2716.
3. Tur, G., U. Guz, and D. Hakkani-Tur, *Model Adaptation for Dialog Act Tagging*. Proceeding of the *IEEE/ACL Workshop on Spoken Language Technology*. 2006: Palm Beach, Aruba. p. 94-97.
4. Ju, Y.-C., Y.-Y. Wang, and A. Acero, *Call Analysis with Classification Using Speech and Non-Speech Features*. Proceedings of *INTERSPEECH*. 2006: Pittsburgh, Pennsylvania.
5. Yu, D., et al., *Automated Directory Assistance System - from Theory to Practice*. Proceedings of *INTERSPEECH*. 2007: Antwerp, Belgium. p. 2709-2712.
6. Berger, A.L., S.A.D. Pietra, and V.J.D. Pietra, *A Maximum Entropy Approach to Natural Language Processing.* Computational Linguistics, 1996. **22**(1): p. 39-72.
7. Kushner, H.J. and G.G. Yin, *Stochastic Approximation Algorithms and Applications*. 1997: Springer-Verlag.
8. Robertson, S., *Understanding inverse document frequency: on theoretical arguments for IDF.* Journal of Documentation 2004. **60**(5): p. 503 - 520
9. Wang, Y.-Y. and A. Acero, *Maximum Entropy Model Parameterization with Tf*Idf Weighted Vector Space Model*. Proceedings of the *IEEE Workshop on Automatic Speech Recognition and Understanding*. 2007: Kyoto, Japan.
10. Brown, R.D., *Example-Based Machine Translation in the Pangloss System*. Proceedings of *the 16th International Conference on Computational Linguistics (COLING-96)*. 1996: Copenhagen, Denmark. p. 169-174.
11. Li, X., Y.-Y. Wang, and A. Acero, *Learning Query Intent from Regularized Click Graphs*. Proceedings of *the 31st Annual International ACM SIGIR Conference*. 2008.
12. Alexandrescu, A. and k. Kirchhoff. *Graph-Based Learning for Phonetic Classification*. Proceeding of the *IEEE Workshop on Automatic Speech Recognition and Understanding*. 2007. Kyoto, Japan.