Evaluation of Background Knowledge for Latent Semantic Indexing Classification

Sarah Zelikovitz
College of Staten Island of CUNY
2800 Victory Blvd
Staten Island, NY 10314
zelikovitz@mail.csi.cuny.edu

Finella Marquez
College of Staten Island of CUNY
2800 Victory Blvd
Staten Island, NY 10314

Abstract

This paper presents work that evaluates background knowledge for use in improving accuracy for text classification using Latent Semantic Indexing (LSI). LSI's singular value decomposition process can be performed on a combination of training data and background knowledge. Intuitively, the closer the background knowledge is to the classification task, the more helpful it will be in terms of creating a reduced space that will be effective in performing classification. Using a variety of data sets, we evaluate sets of background knowledge in terms of how close they are to training data, and in terms of how much they improve classification.

Introduction

Text Classification and Unsupervised Learning

Categorizing textual data has many practical applications, including email routing, news filtering, topic spotting, and classification of technical documents. Traditional machine learning programs use a training corpus of hand-labeled training data to classify new unlabeled test examples. Often the training sets are extremely small, due to limited availability of data or to the difficult and tedious nature of labeling, and classification decisions can therefore be difficult to make with high confidence. Other sources of information that are related to the task often exist, and it is important for text classifiers to take advantage of these additional resources. This information can be looked at as background knowledge that can aid in the classification task.

Latent Semantic Indexing

One method of incorporating background knowledge in a nearest neighbor paradigm, uses a latent semantic indexing (Deerwester *et al.* 1990; Dumais 1996) (LSI) approach (Zelikovitz & Hirsh 2001). LSI creates a matrix of documents, and uses singular value decomposition to reduce this space to one that hopefully reflects the relationships between words in the textual domain. The addition of background knowledge into this matrix allows for the decomposition to reflect relationships of words in the background knowledge as well. However, in order for the additional knowledge to be useful for classification it must be related to the text categorization task and to the training data. In the extreme case, if many pieces of background knowledge are added to the training examples, and if the background knowledge is unrelated to the task, we can imagine that LSI would create a space that simply reflects the irrelevant background knowledge. This new space would be essentially useless for the classification of unseen examples.

Understanding Background Knowledge

Previous research has shown that in many text categorization problems, the addition of longer, relevant textual background data allows for richer co-occurrences to be modeled properly.

We can think of evaluating background knowledge in terms of how closely it is related to the categorization task. Unlabeled examples, that come from the same source and time period as the set of training and test examples, would be most appropriate in terms of the concepts and vocabulary that we wish to model during classification. Background knowledge created via web searches might be less suitable. However, for some text classification tasks, unlabeled examples might not be the best form of background knowledge. One such task is the classification of business names by the industry to which it belongs. For example, *Addison Wesley* would be classified as *publisher*. In this case the names are too short to give much information about the task. However, other data which might not even be classifiable, such as business news articles, or company web pages might be more helpful in finding useful word relationships. An important research question that we address is "how close must the background knowledge be to the task in order to be useful?"

We take the approach that to evaluate the closeness of a set of background knowledge to a classification task, we much essentially measure the relationship between the background set and the training set. Since our final classification method of LSI is a nearest neighbor approach, where the unlabeled test examples are compared with the original labeled training examples (in the new space), a set of background knowledge can only be helpful if it allows the training examples to be modeled in some better way. Hence, we assess our set of background knowledge by determining how close it is to the training set.

Measurements

A document from the training or background set, x_j, is represented as a vector of terms weights $\langle w_{j1}, \ldots, w_{j|T|} \rangle$, where T is the set of terms in all the documents. These weights are generated using the standard TFIDF method (Salton 1989), where w_{jt} equals $log(TF_{x_jt}) \times log(IDF_t)$. TF_{x_jt} corresponds to the number of times that term t occurs in document x_j and IDF is the total number of documents divided by the number of documents that contain the term t. Two document vectors, x_i and x_j are compared using the cosine similarity metric. Since all vectors are made to be of unit length this value is always between 0 and 1.

We measure the cosine similarity of each piece of background knowledge with each training example, and sort these pairs in descending order by similarity. This list is then used to determine the following two threshold values.

Firstly, we traverse this list until we have seen 1% of the background knowledge set, and report the cosine similarity at that line. If many of the pieces of background knowledge are close to the training data, we would expect this number to be close to the closest similarity metric. On the other hand, if only a very few pieces of background knowledge are close to the training data, we might have to traverse the similarity of those few close examples with all the training before seeing 1% of the background set. We continue this traversal for each percentage of the background set, ranging 1-100, reporting the cosine value each time that percentage of the background set has been seen.

A second method of viewing this sorted sequence of pairs of background and training examples is by determining what percentage of *training* examples have been seen when 1% of the background set has been seen. If there are many training examples that are similar to many of the background set examples we would expect this number to be high. If very few training examples are very similar to the entire background set, then this number would be low. We can get his value for each percentage (1%-100%) of the background data.

Creating Subsets of Background Knowledge

Our experiemental tests were performed on three data sets/ background knowledge sets that we and other researchers used for text classification (Cohen & Hirsh 1998; Zelikovitz & Hirsh 2001).

We can use our measurements of background knowledge from the section above to create multiple sets of background knowledge from our full background knowledge set. Specifically, following method one, we created 10 sets of background knowledge from the original full set. The first set contained the 10% closest pieces of background knowledge to the training set, the next set contained the closest 20%, the next set contained the closest 30%, etc. The last set contained the full background knowledge set. We used each of these sets with different sized training sets, in our three test domains. For two of our domains, for larger data sets, almost all of the gain in accuracy takes place when only the 50% closest pieces of background knowledge are used. Additional background knowledge does not improve classification. For the smaller data sets, up to about 75% of the background was necessary before full improvement in accuracy was reached. This is expected because more background knowledge is needed to compensate for the lack of vocabulary and cooccurrences of words in the smaller training corpora. Our third set exibited drastically different behavior. Here, the gain from the background knowledge is only seen once almost the entire set of background knowledge is used. This phenomenon can be understood by looking at the second method of creating subsets of background knowledge.

For this second method, we created 10 sets of background knowledge as well. The first set contained all those pieces of background knowledge that were the closest to the training data, once 10% of the training data was seen in the sorted similarity list. The second set contained those pieces that were closest to 20% of the training set, etc.

The important point that we learned from evaluating these background sets in conjunction with the LSI nearest neighbor learner, is that for all the three test domains the major increase in accuracy is at the point when enough background knowledge is included, so that *many* of the training examples are close to pieces of background in the set. Even if many background pieces were included, if they were close to a few of the training examples, the accuracy results did not improve, simply because much background was added. However, if only a few background pieces were added but were close to many of the training examples, then improvement was noticeable.

Conclusion

We presented a method for comparing a background knowledge set to training examples for use with LSI classification. Our approach compared individual pieces of background knowledge with individual training examples. We conclude from our text classification experiments that to be most useful in the singular value decomposition process, a background set should contain data that is similar to much of the training set.

References

Cohen, W., and Hirsh, H. 1998. Joins that generalize: Text categorization using WHIRL. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 169–173.

Deerwester, S.; Dumais, S.; Furnas, G.; and Landauer, T. 1990. Indexing by latent semantic analysis. *Journal for the American Society for Information Science* 41(6):391–407.

Dumais, S. 1996. Combining evidence for effective information filtering. In *AAAI Spring Symposium on Machine Learning and Information Retrieval, Tech Report SS-96-07*.

Salton, G., ed. 1989. *Automatic Text Processing*. Reading, Massachusetts: Addison Welsley.

Zelikovitz, S., and Hirsh, H. 2001. Using LSI for text classification in the presence of background text. In *Proceedings of the Tenth Conference for Information and Knowledge Management*, 113–118.