LARGE-SCALE WORD REPRESENTATION FEATURES FOR IMPROVED SPOKEN LANGUAGE UNDERSTANDING

Jun Zhang *Terry Zhenrong Yang* *Timothy J. Hazen*

Microsoft Corporation
New England Research and Development Center
Cambridge, Massachusetts, USA

ABSTRACT

Recently there has been great interest in the application of word representation techniques to various natural language processing (NLP) scenarios. Word representation features from techniques such as Brown clustering or spectral clustering are generally computed from large corpora of unlabeled data in a completely unsupervised manner. These features can then be directly included as supplementary features to standard representations used for NLP processing tasks. In this paper, we apply these techniques to the tasks of domain classification and intent detection in a spoken language understanding (SLU) system. In experiments in a personal assistant domain, features derived from both Brown clustering and spectral clustering techniques improved the performance of all models in our experiments and the combination of both techniques yielded additional improvements.

Index Terms— word representation, hierarchical clustering, spectral clustering, spoken language understanding

1. INTRODUCTION

Spoken language understanding (SLU) systems are now widely available and commonly used in a variety of application areas. For example, mobile phone personal assistants such as Siri, Google Now, and Cortana provide users with a wide range of functionality for controlling their devices and accessing information services. Common tasks handled by these systems include setting alarms, handling mobile communications, managing calendar entries, checking weather forecasts, and finding nearby restaurants. These mobile based systems are typically supported by a collection of online services that perform computational tasks such as speech recognition and spoken language understanding and provide access to question answering or web search services.

In this work, we focus on new methods for improving the SLU modeling used in such personal assistant systems. Our SLU system employs the commonly used domain/intent/slot modeling approach [1]. The system architecture consists of three cascaded SLU components: (1) a domain classifier, (2) a user intent classifier, and (3) a semantic slot tagger. Domain classification and intent classification are performed using support vector machines (SVMs) that operate on high-dimension feature vectors extracted from user queries. The feature vectors contain a rich collection of features including word-based n-gram features and dictionary-based features indicating the presence of words or phrases contained in different lexical dictionaries. The slot tagging component also uses a similarly rich high-dimension collection of features to determine semantic slot tags for individual words or word sequences in the query using techniques such as conditional random fields (CRFs) or recurrent neural networks (RNNs).

The SLU approach described above applies supervised training techniques that generally require large volumes of annotated data. If human annotators are required, the training data set sizes can be smaller than desired due to limited annotation resources. As a result, a common problem is data sparsity. On the other hand, commercial SLU systems with large user bases can provide access to massive collections of unannotated queries that can be exploited for the discovery of new features learned from unsupervised methods.

Recently, new methods have emerged to address data sparsity problems through the unsupervised learning of word representation features from unlabeled data. The inclusion of these new features into existing modeling approaches has resulted in performance improvements in numerous studies [2, 3, 4, 5], and particularly for the tasks of name entity recognition [6, 7, 8, 9] and syntactic or semantic parsing. A common and successful word representation technique in these studies is based on the Brown clustering algorithm [10]. However, this learning algorithm is computationally expensive. Recently spectral based algorithms [11, 12, 13, 14, 15] have been proposed for fast computation of hierarchical word clustering. These approaches generally first compute eigenvectors from correlation matrices learned from word statistics to provide a word embedding representation. Next, clustering algorithms are applied to the eigenvectors to infer the final hierarchical clustering. Generally these low-dimension word embeddings describe the common latent semantic structure of the words.

In this work, we examine approaches for incorporating

features derived from both Brown clustering and spectral techniques into the domain and intent classification components of our SLU system. We have observed that by simply including word representation features into the feature vectors of our standard classification approach, improvements in classification accuracy are observed from both the Brown clustering techniques and the spectral techniques. Furthermore, using both sets of features in combination achieved even further improvements.

The rest of the paper is organized as follows: in section 2 we briefly review the word representation methods. In subsection 2.1, we describe the Brown cluster features. In subsection 2.2, we describe the spectral based methods, with a focus on canonical correlation analysis (CCA). In section 3, we describe the details of our experimental set-up. We present the results in Section 4 and provide conclusions in Section 5.

2. BACKGROUND

Word representations generally fall into two categories: hierarchical clustering based word representations such as Brown clustering and dense representations. The dense representations are real-valued and low-dimensional, where the coordinates generally represent some hidden states of the syntactic and semantic word properties. There are a few approaches for inferring the embedding such as neural network, log-linear models or spectral based methods.

2.1. Brown Clustering

Brown clustering is a bottom-up algorithm that derives a hierarchical clustering of words. Clustering is performed in a greedy fashion such that the two word clusters with largest mutual information, as measured from bigram context statistics, are merged at each step. The cluster tree yields a hard clustering such that each word belongs to only one branch of the cluster tree. Different depths of the clustering tree represent different levels of semantic similarity with greater generalization near the root of the tree. By allowing the learning algorithm to draw analogies between different words at different levels of the tree, the Brown clustering alleviates the sparsity problem.

2.2. Spectral Based Clustering

Spectral clustering using canonical correlation analysis (CCA) provides an alternative method for word clustering [7, 11, 12]. CCA is a statistical analog of principal component analysis (PCA) and can compute the projection of maximal correlation between pairs of matrices. When applied to bigram context distributions of words, the top few eigenvectors of CCA with the highest eigenvalues can approximate the latent structure of words. After these eigenvectors are computed, clustering algorithms such as K-means can be applied to infer word clusters. Recently the work by Hsu et al [14] suggests that CCA type spectral algorithms can learn a Brown cluster model in polynomial sample/time complexity. Stratos et al [15] also proposed a new CCA based approach using a bottom-up agglomerative clustering algorithm. We follow their approach for the word embedding computation in this paper.

3. EXPERIMENTAL SETTING

In the experiments carried out below, we use SVMs as supervised classifiers for both the domain classification and intent detection. There is some experiment design difference between the two tasks. In our domain classification task, we first build a SVM based binary classifier for each domain and transform the output via a sigmoid function [16] to a score, and then compare the scores across all domains. Intent detection classification is performed in the same fashion, but is carried out with domain specific intent models inside each respective domain. Note within each domain, the number of intents varies from 8 to 20. So each domain specific intent classifier is a multi-class SVM, which differs from the binary SVM for each domain model. For each task, we compute the SVM with a basic feature set as the baseline. For our experimental comparison, we then add Brown cluster (BC) features, word embedding (WE) features, and both Brown cluster and word embedding features (BC+WE) to the baseline features.

3.1. Data

The domain and intent classification experiments in spoken language understanding are run in seven domains. These domains generally refer to the categories of service quests. For example, the query ”how is the weather at LA?” is mapped to the weather domain. If the query falls outside of the seven domains, it is assigned to the ”other” domain (Domain 8 in tables below). Our data set is divided into three parts: training, validation and test. The training data has approximately 540K spoken queries, and the validation and test sets each contain approximately 27K queries.

3.1.1. Unlabeled data

Independent of the datasets used in the train/validation/test procedures, we randomly sampled 24 million unannotated spoken queries collected by a preexisting personal assistant system, and used these for computing the word representation features. The dataset contains approximately 102 million tokens and one million unique words. We applied the same preprocessing procedures, such as routine text normalization, to these unlabeled data as to the training/validation/test data. We used the C++ implementation of Brown clustering by Liang [3]. As observed by many others, the algorithm is

computationally expensive. In our experiment, it took approximately 80 hours on a multi-threaded 4-core machine to produce the Brown clustering for the full data set. On the other hand the CCA based spectral method is much faster, typically finishing the same computation within an hour.

3.2. Baseline Feature Sets and Model

We used all of the word unigrams, bigrams and trigrams in the training datasets as baseline features. The dimensions of them are respectively 92K, 643K and 1,208K with total 1,945K features. Although our task has domain specific class based lexicon dictionaries available, we discovered features derived from these lexicons did not improve performance beyond the basic word ngram features, and hence are not used in these experiments. Both the domain classification and user intent detection use the multi-class SVM classifier algorithm formulated by Cramer and Singer [17]. For the purpose of fast training on the large dataset, we restrict the SVM to the simple linear kernel, as the nonlinear kernels require a significant increase in training time due to the very high dimension of ngram features in our setting. The implementation is similar to the widely used package Liblinear [18].

3.3. Including Word Cluster Features

To use the word cluster features, we simply add indicator features which equal one when a word in the utterance appears in the corresponding cluster. This is equivalent to a unigram indicator function over the sequence of Brown cluster labels. Words that are unseen in the unlabeled dataset are assigned to the special cluster of UNK, as is common in the literature. As the Brown clusters are essentially hierarchical trees, different depths represent clustering at different levels of generality. One can extract clustering features at multiple depths of the hierarchy. Following Ratinov & Roth [9], we used clusters at depth 4, 6, 10 and 20. Though the features generated at different levels overlap, generally the learning algorithm will automatically select those most useful features.

3.4. Including Word Embedding Features

In our experiments we use word embedding features derived from CCA as described by Stratos [15]. Word embedding features map each word to a real-valued vector of fixed dimension d, where d is selected in advance. The word embedding feature vector for a query is the component-wise average of all the associated vectors of the individual words from the query. So the word embedding feature dimension increases by only d. The word embedding features are continuous and real-valued. They differ from the other binary features in our feature sets which only takes 0 or 1 as their values. To include them as extra features together with other binary features, it is helpful to suitably scale them for best performance. The model can be sensitive to the scaling, as observed by others in the literature [19, 3]. Following Turian et al [19], we assume the embedding matrix is denoted as a matrix E, where E_i is the row for the i^{th} word in the vocabulary. We use the overall scaling strategy by normalizing E_i with constant $Max(E_i)$. In the experiments we tried the dimension to be $50, 100, 150, 200, 250$ and 300. We stopped at 300 due to memory limitations, as our vocabulary size goes beyond 100k.

4. RESULT AND DISCUSSION

We reported results in the Tables 1 and 2. We denote Brown cluster and word embedding feature sets respectively by BC and WE. The Baseline models simply use all the unigram, bigram and trigram features in the training data sets. For the domain model result in Table 1, we observed that including Brown cluster features achieved a significant error drop for most domains. On average across all eight domains the error rate dropped from 5.51% to 5.31%. For WE features, the error rate reduces to 5.37%. When we combined Brown cluster and word embedding features, the average error rate of the domain model dropped from 5.51% to 5.25% (a relative 4.7% drop).

In the intent model in Table 2, all seven domains achieved a reduction in error rate for the BC features. The average error rate across seven domains drops from 5.20% to 4.65% (a relative 10.3% drop) for BC features, and to 4.74% (relative 8.69% drop) for WE features. When BC and WE features are both included, we still observe slight error reduction with error rate drops to 4.60%. Compared with the domain model result in Table 1, the effect of combining both BC and WE is small. We hypothesize that this may be due to the design difference of the experiment procedures. Generally both BC and WE are respectively discrete (one-hot) and continuous (compact) word representations. Note they essentially assume the same bigram Brown cluster model. For the word embedding features, we observed a similar result with slightly worse performance compared with Brown cluster features. This observation is consistent with other reports in the literature [19, 15]. A general hypothesis is that BC features works better with linear classifiers than the real-valued WE features. In our experiments, we used a corpus count threshold of 20 for inclusion of words in the computation of word embedding. In the Table 1 and 2, the word embedding feature vectors have a dimension of 300. We observed similar performance for dimensions from 100 to 300.

We also tried including the bigrams and trigrams of Brown cluster features in the model. For example, a query of three words $W_1W_2W_3$ where each word W_i has corresponding class C_i. Then the BC bigrams are C_1C_2 and C_2C_3 and the BC trigram is $C_1C_2C_3$. However, these features do not improve the model accuracy further.

Domain	Baseline	+BC	+WE	+BC+WE
D1	10.84%	10.10%	10.22%	9.95%
D2	4.45%	4.12%	4.30%	4.05%
D3	10.07%	8.97%	9.14%	8.65%
D4	5.96%	5.92%	5.92%	5.88%
D5	4.29%	4.33%	4.28%	4.29%
D6	2.30%	2.25%	2.28%	2.20%
D7	4.81%	5.54%	5.53%	5.45%
D8	4.22%	4.21%	4.21%	4.20%
Average	5.51%	5.31%	5.37%	5.25%

Table 1. Domain Model Error Result. Baseline stands for the base model with only n-gram features. BC stands for Brown cluster feature. WE stands for word embedding features of dimension 300.

Domain	Baseline	+BC	+WE	+BC+WE
D1	5.96%	5.38%	5.46%	5.38%
D2	5.72%	4.68%	4.98%	4.81%
D3	7.96%	7.29%	7.41%	7.24%
D4	5.12%	5.19%	4.75%	4.82%
D5	1.43%	1.29%	1.52%	1.15%
D6	4.73%	4.32%	4.42%	4.23%
D7	5.41%	4.53%	4.61%	4.53%
Average	5.20%	4.64%	4.75%	4.60%

Table 2. Intent Model Error Result. Baseline stands for the base model with only n-gram features. BC stands for Brown cluster feature. WE stands for word embedding features of dimension 300.

5. CONCLUSION

In this paper, we provided empirical result of using word representation techniques to the classification tasks in SLU. We computed word representation features on a large volume of unlabelled data in a real SLU system, and included them as additional features in the SVM based classifiers for domain classification and intent detection tasks in SLU. Significant error rates drop have been observed. For the spectral based clustering algorithm, we proposed directly using the word embedding features rather than the inferred clusters. For the optimal usage of unlabelled data, we found combining the classical Brown features and word embedding features gave the best performance.

6. ACKNOWLEDGMENT

The authors would like thank Sham Kakade for fruitful discussion and guidance on the topics of the paper.

7. REFERENCES

[1] Y.-Y. Wang, L. Deng, and A. Acero, "Semantic frame-based spoken language understanding", in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, G. Tur and R. De Mori, editors, John Wiley and Sons, Chichester, UK, 2011.

[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[3] P Liang, "Semi-supervised learning for natural language," *Masters thesis*, Massachusetts Institute of Technology, 2005.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111-3119.

[5] T. Mikolov W.-T. Yih and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. of NAACL-HLT*, 2013, pp. 746-751.

[6] S. Miller, J. Guinness and A. Zamanian, "Name tagging with word clusters and discriminative training," in *Proc. of HLT-NAACL*, 2004, pp. 337-342.

[7] R. Sarikaya, A. Celikyilmaz, A. Deoras and M. Jeong, "Shrinkage based features for slot tagging with conditional random fields," in *Proc. Interspeech*, 2014.

[8] J. Turian, L. Ratinov and Y. Bengio, "Word representations: a simple and general method for semisupervised learning," in *Proc. of Annualt meeting of ACL*, 2010, pp. 384–394.

[9] L. Ratinov and D Roth, "Design challenges and misconceptions in named entity recognition," in *Proc. of Conf. on Computational Natural Language Learning*, 2009, pp. 147-155.

[10] P. Brown, P. deSouza, R. Mercer, V. Della Pietra and J. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467-479, 1992.

[11] A. Ng, M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Information Processing Systems Conf. (NIPS)*, 2002.

[12] P. Dhillon, D. Foster and L. Ungar, "Multi-view learning of word embeddings via CCA," in *Proc. Neural Information Processing Systems Conf. (NIPS)*, 2011.

[13] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16 no. 12, pp. 2639-2664, 2004.

[14] D. Hsu, S. Kakade and T. Zhang, "A spectral algorithm for learning hidden markov models," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1460-1480, 2012.

[15] K. Stratos, D.-K. Kim, M. Collins and D. Hsu, "A spectral algorithm for learning class-based n-gram models of natural language," in *Proceedings of UAI*, 2014.

[16] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in large margin classifiers*, pp. 61–74, MIT Press, 1999.

[17] Koby Crammer and Yoram Singer, "On the learnability and desgin of output codes for multiclass problems," *Computational Learning Theory*, pp. 35–46, 2000.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[19] J. Turian, L. Ratinov, Y. Bengio and Dan Roth, "A preliminary evaluation of word representations for named-entity recognition," in *Proc. NIPS Workshop on Grammar Induction, Representation of Languauge and Language Learning*, 2009.