Shrinkage Based Features for Slot Tagging with Conditional Random Fields

Ruhi Sarikaya, Asli Celikyilmaz, Anoop Deoras, Minwoo Jeong

Microsoft Corporation, Redmond, WA 98052

{ruhi.sarikaya, asli.celikyilmaz, anoop.deoras, minwoo.jeong}@microsoft.com

Abstract

In this paper we propose a set of class-based features that are generated in an unsupervised fashion to improve slot tagging with Conditional Random Fields (CRFs). The feature generation is based on the idea behind shrinkage based language models, where *shrinking* the sum of parameter magnitudes in an exponential model tends to improve performance. We use these features with CRFs and show that they consistently improve the slot tagging performance against baselines on several natural language understanding tasks. Since the proposed features are generated in an unsupervised manner without significant computational overhead, the improvements in performance comes for free and we expect that the same features may result in gains in other tagging tasks.

Index Terms: shrinkage based features, CRFs, exponential models, slot tagging

1. Introduction

Recent releases of digital personal assistants such as Siri, Google Now and Dragon Go in smart phones made Spoken Language Understanding (SLU) more important than ever in completing routine tasks (e.g. finding a restaurant, checking weather, checking flight status, setting up a meeting in calendar) using voice in our daily lives. Most of these tasks involve extracting information from the spoken query. The information could be at the sentence level (e.g. domain detection, user intent detection) or at the word/phrase level (e.g. semantic concepts, entities). The latter combined with the former provides a granular understanding of the user's goal and also allows formulating queries to fetch information from the knowledge back-end for these applications.

Despite continuous research over the past two decades, semantic information extraction in the form of slot filling in SLU is still a challenging problem. Even though representation learning methods such as deep learning are starting to receive attention [3, 4, 14, 9], Conditional Random Fields (CRFs) [2] have been the most widely used technique for slot filling. CRFs benefit from the flexibility to use overlapping, non-independent features to model many natural language tasks. This can reduce the need for labeled data by taking advantage of domain knowledge in the form of dictionaries, part-of-speech tags, syntactic parsing, and capitalization patterns. Given the great flexibility of these models to use a wide variety of features, two important questions remain 1) what features to use, 2) whether the runtime computational requirements justify the accuracy gains with the extra features given the system handles potentially millions of queries every day. It is also important to know whether the extra features require hand-labeling of the data, which could be an expensive task and that the performance gains may not justify generating such annotations.

Until recently there has not been a principled approach to determine if a given feature (e.g. lexical, syntactic, semantic) provides gains in accuracy for exponential models without actually building and evaluating the models. In [6] [7] it was empirically shown that for maximum entropy based language models, which are specific instances of exponential models, the performance could be predicted for features of the same kind (e.g. lexical n–grams) under certain assumptions. In a later study [1], it was shown that automatically derived class based features could shrink the model size and lead to performance gains. One of the key outcome of the study was that the features are derived in an unsupervised fashion and thus making them potentially applicable to other tasks (beyond language modeling) and languages.

This paper takes an empirical approach to transfer the findings of [1, 6, 7, 11, 16] to the slot filling task in SLU. We propose a set of automatically generated feature sets for slot tagging using CRF modeling. We use (hard and soft) word clustering to induce class based features. We consider two word clustering techniques for hard clustering: Brown clustering and spectral clustering and their combination for soft clustering.

This paper is organized as follows: Section 2 provides a formulation of conditional random fields. Section 3 describes the empirical basis of the shrinkage based features. Section 4 introduces the shrinkage based features for slot filling. Section 5 presents experimental results followed by conclusions in Section 6.

2. Conditional Random Fields

Conditional Random Fields (CRFs) [2] are discriminative undirected probabilistic graphical models trained to maximize the conditional probability of labels on output nodes given the observations on the input nodes. If the output nodes of the graphical model are linked by edges in a linear chain, CRFs make a first-order Markov independence assumption, and thus can be understood as conditionally-trained finite state machines (FSMs). Unlike ordinary classifiers, which predict a label for a single sample without considering the neighboring samples, a CRF can take (label) context into account and models sequences of labels. For example, a linear chain CRF predicts sequences of slots for sequences of input samples (i.e. words). Assuming that $n = 1$ is the length of the observation sequence, a linear-chain CRF can be written as:

$$p_\lambda(\mathrm{y} \mid \mathrm{x}) = \frac{1}{Z_\lambda(\mathrm{x})} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \mathrm{x}, j\right)\right) \quad (1)$$

where j denotes the position in the input observation sequence $\mathrm{x} = \{x_1, \ldots, x_m\}$ and $\mathrm{y} = \{y_1, \ldots, y_m\}$ is the output sequence. $f_i(.)$ are often binary valued (but can be real-valued as well) feature functions, which depend both on the input observation sequence and output label sequence. Model parameters (λ_i) are learned weights associated with feature $f_i(.)$ and they

are independent of the position j. $Z_\lambda(\mathrm{x})$ is the normalization term to make sure the expression is a probability:

$$Z_\lambda(\mathrm{x}) = \sum_{y \in Y} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \mathrm{x}, j\right)\right) \quad (2)$$

where summation over Y, the set of all possible label sequences, makes the probabilities sum to one. Within the $\exp(.)$ function, we sum over $j = 1, \ldots, n$ word positions in the sequence. Given such a model the most likely label sequence for an input sequence x is,

$$y^* = \arg\max_y P_\Lambda\left(\mathrm{y} \mid \mathrm{x}\right) \quad (3)$$

where $\lambda_i \in \Lambda$, $i = 1 \ldots m$. This expression can be efficiently computed using the Viterbi algorithm. Being from the same exponential family of model CRFs share many of the properties of standard maximum entropy models, including their convex likelihood function, which guarantees that the learning procedure converges to the global maximum. Traditional maximum entropy learning algorithms, such as GIS and IIS [8], can be used to train CRFs. However it is widely known that a stochastic gradient descent (SGD) converges much faster compared to GIS or IIS [13], so we use SGD for learning the model parameters.

3. The Empirical Basis for Shrinkage Based Exponential Models

Shrinkage based language models are class-based exponential n-gram models that achieved significant gains over word-based n-gram models on a number of speech recognition and machine translation tasks [1, 11, 12]. The basic premise of shrinkage-based exponential language models lies in *shrinking* the model size of the word n-gram models. Assuming that training and test data are drawn from the same distribution, it was shown empirically [1, 6] that many types of shrinkage based exponential language models obey the following rule:

$$\log \mathrm{PP}_{\mathrm{test}} \approx \log \mathrm{PP}_{\mathrm{train}} + \frac{\gamma}{D}\sum_i |\tilde{\lambda}_i| \quad (4)$$

where $\mathrm{PP}_{\mathrm{test}}$ and $\mathrm{PP}_{\mathrm{train}}$ denote test and training set perplexity; D is the number of words in the training data; $\tilde{\lambda}_i$ are *regularized* (i.e., smoothed) estimates of the model parameters; and γ is a constant independent of domain, # utterances in training data, and model type [6].

The above relationship suggests that test set performance can be improved if we shrink $\sum_i |\tilde{\lambda}_i|$ (i.e., *model size*) while maintaining the training set performance. There are various ways to shrink the model size such as through regularization during model training. In fact such techniques are widely used in training exponential models. Another empirical observation described in [6, 7] suggests alternative ways to shrink the model size, and it may complement the regularization during model training. We observe that when building an exponential model such as language model or a general classification model [16], whenever back-off features are added, the model performance improves. For example, adding bigram features to a model trained with trigram only features, or adding unigram features to a model trained with bigram only features improves the model performance. It was shown in [6, 7] that these back-off features shrink the model size, which is the second term in Eqn. 4. Note that adding bigram features to a model trained with trigram only features increases the number of model parameters (i.e. $\lambda's$) but sum of the absolute values of them decreases effectively *shrinking* the overall model size. When bigram features are added to trigram features and a new model is trained with both feature sets, the absolute values of the model parameters corresponding to trigram features shrink even though the model has more parameters to learn (i.e. bigram+trigram). Since the shrinkage in absolute value of trigram model parameters is larger than the absolute values of the new parameters learned for the bigram features and there is a net reduction (i.e. shrinkage) in the model size. In the experiments, we will show analysis of shrinkage on model performance.

F-1gr	$c_j, w_j, c_j w_j$
Feature SetA (F-A)	$c_j, c_{j-1}c_j, w_{j-1}c_j, w_j, c_j w_j, w_{j-1}w_j$
Feature SetB (F-B)	$c_j, c_{j-1}c_j, w_{j-1}c_j, w_j, w_{j-1}c_j w_j, c_j w_j$
Feature SetC (F-C)	$c_j, c_{j-1}c_j, w_j, c_j w_j, w_{j-1}w_j$
Feature SetD (F-D)	$c_j, c_{j-1}c_j, w_j, c_j w_j, w_{j-1}w_j c_j$
Feature SetE (F-E)	$c_j, c_{j+1}, c_j w_j, w_{j+1}, w_{j+1}c_{j+1}$

Table 1: Different sets of shrinkage-based features. Note that 2gr features include the 1gr features as a subset (Feature set A through E).

The empirical observations given above and the relationship in Eqn. 4 helps to explain why we get improvements when we add bigram features to trigram features or when we add unigram features to both bigram and trigram features. Given these observations a set of class based exponential language models are proposed in [1, 6, 11] with the goal of shrinking the model size while maintaining the training set performance. These language models lead to significant improvements in test set performance over the state-of-the-art language models. In the experiments, we will show analysis of shrinkage on model performance.

4. Shrinkage Based Features for CRF

In this study we are leveraging the same empirical evidence to propose class based features for CRFs, which belong to the same exponential family of models. We propose several feature sets described in Table 1. We use CRFs for slot filling task. In the baseline setup, we consider 1-gram and 2-gram word features within a window of 5 words (± 2 words) around the current word as binary feature functions. The features are in general joint lexical and class based features composed of 1grams and 2grams. In the table w_j is the jth word in the sentence and c_j is its corresponding class. Word classes can be generated through supervised or unsupervised techniques. Each word can be assigned to a single class (i.e. hard word classing) or multiple classes (soft classing) [11]. The two hard word clustering techniques we use for assigning word classes to the words are:

4.1. Brown Clustering

Brown hierarchical word clustering algorithm [5] partitions the vocabulary into a pre-defined number of classes to maximize the bigram mutual information between words and classes [5]. The algorithm first assigns each of the most frequent words to their own class and the remaining words to the final class. Then, the *exchange* algorithm is performed where individual words are moved to another class if this improves the class bigram mutual information, until no more such moves are possible. In this paper, we use the C++ implementation of Brown clustering [1].

[1] https://github.com/percyliang/brown-cluster

4.2. Spectral Clustering

We also form word classes using spectral clustering [17] of word embeddings. A word embedding is a continuous representation of a word. In this work, we use off the shelf word embeddings described in [18, 19] on large amounts of newswire data (which, for our study, is completely out of domain data). The embeddings are derived for 268K words and each word is represented by a 50 dimensional vector of continuous values.

In order to use these word embeddings in our shrinkage based CRF framework, we generated word classes out of these continuous word representations. Although K-means clustering seems an obvious choice for forming clustering, we however used spectral clustering [17], mainly because of spectral clustering's superiority over K-means in forming clusters of non-spherical shapes. We follow the procedure described in [17] to form spectral clustering of 268K words. However, instead of using a fully connected affinity graph, which would have a prohibitively large number of edges (6.7×10^{10}), we use K nearest neighbors (symmetric version, with K=5) and obtain spectral decomposition of the resulting sparse graph (which contains about 5×10^5 edges maximum) using sparse Eigen solvers.

4.3. Soft Clustering

The soft clustering is often desired, as it encodes uncertainties on data-to-cluster assignments. A single observation may often belong to more than one cluster, e.g., a document with multiple themes may belong to different topics. In this paper, we approach soft clustering by using the features from Brown and spectral clustering. Specifically, we assume that a words and thus the *n*-gram features may belong to more than one cluster indicated by different clustering algorithms. Thus, each word in the training data is assigned to two separate clusters, one from Brown clusters and one spectral clustering. The features generated separately and combined during model training and testing.

5. Experimental Results

5.1. Data

The slot filling experiments are run on three tasks covering entertainment search scenarios: 1) movie domain, 2) music domain, 3) games domain. The statistics about the datasets used in this paper are given in Table 2.

	Tag Count	Train # Utterances	Test # Utterances
Movie	33	32147	10695
Music	20	7513	2499
Games	17	7523	2500

Table 2: Data Set Descriptions.

5.2. CRF Model Performance with Shrinkage Features

We trained the CRF models using the feature sets shown in Table 1 for cluster sizes starting from 50 up to 500 with 50 increments on each domain data. We measured the performance of the models on the test data using the F-score. We present the results and analysis of the effects of using different clustering methods and features sets separately below.

5.2.1. Cluster Size vs. Clustering Technique

Here, we investigate whether the cluster size and clustering methods has effect on the model performance. Fig. 1 shows the results where we build separate CRF models using Brown, Spectral and soft clustering (Brown+spectral) with the defined

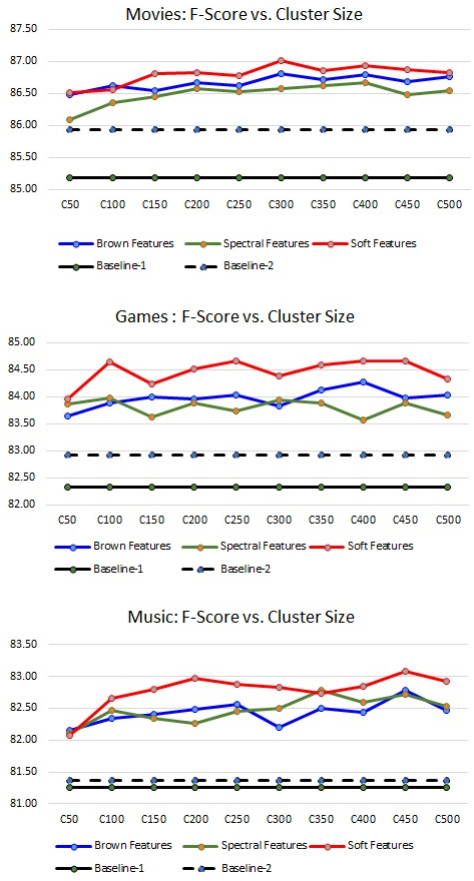

Figure 1: Slot tagging performance for (top) Movies, (middle) Games, and (bottom) Music domain test sets **for different cluster sizes** of clustering features. Each line on each chart indicates the average model F-scores of the feature sets when (∼) Baseline-1 with unigram features, (doted-∼) Baseline-2 with unigram+bigram features, (∼) Brown, (∼) Spectral, and (∼) Soft clustering are used.

features sets for a set of fixed cluster sizes. Specifically, we take cluster-ids as features (as shown in Table 1) for each feature type and build a separate model for each feature combination per cluster size. Because of space limitations, we cannot show results from each features set combination using each clustering model per domain. Instead, In Fig. 1, we show the *average* F-scores on test sets predicted from the CRF models trained with different feature sets, but using the same cluster size for all the feature sets. Each chart compares Brown, Spectral and Soft word clustering results for each domain. The baseline models use word unigram (1gr) and unigram+bigram (1gr+2gr) features without any clustering features to compare with the clustering based (class) features.

We observe that the feature sets perform similarly with Brown and Spectral clustering, however both of them outperformed the two baselines. In each domain, we observe a slight increase in performance as we use large number of clusters (>250), however, we did not see a statistically significant improvement (based on Student t-test of $p \leq 0.01$). Thus, our analysis show that the tagging performance is not sensitive to the cluster size. The best performance is observed with the Soft clustering (significant for music and games using paired t-test, $p < 0.05$), which combines each individual feature from Brown

Figure 2: Slot tagging performance for (top) Movies, (middle) Games, and (bottom) Music domain test sets **for different features sets**. Each line on each chart indicates the average model F-scores of features with different cluster sizes when (∼) Baseline-1 with unigram features, (doted-∼) Baseline-2 with unigram+bigram features, (∼) Brown, (∼) Spectral, and (∼) Soft clustering are used.

and Spectral clusters.

5.2.2. *Feature Sets vs. Clustering Technique*

In the second set of analysis, our goal is to analyze the effects of different word-class feature sets, and benchmark the performance results against the baseline models trained using only the unigram and unigram+bigram feature sets. The results are shown in Figure 2. Again, we compare the models using features from Brown, spectral and soft word clustering against the two baselines. We observe that the best models are with the feature sets F-A, F-B, F-C and All (union of F-A, F-B, F-C, F-D and F-E) in all domains (movies, games and music). Additionally, even though F-1gr and Feature SetE (F-E) achieved lower accuracy compared to other feature sets, they still outperformed the baseline.

Across all three domains we observe that the shrinkage based features consistently improve the performance over the baselines. We also observe that features obtained from the soft clustering (Brown+spectral) consistently outperform those obtained with hard clustering with Brown or spectral. Even though the performance of F-A to F-D are close to each other, F-B is slightly better overall with soft clustering across the domains. If we pick a single feature set, clustering technique and a cluster size: setting the cluster size to 400 (c400) and using soft clustering, F-C improves the accuracy from 85.2/85.9% (1gr/2gr baselines) to 87.5% in the movie domain. With the same settings the improvements in the music domain are from 81.3/81.4% to 83.3%, and in the games domain from 82.3/82.9% to 85.1%. These are significant (using paired t-test, $p < 0.05$) and consistent gains across different domains.

5.3. Shrinkage Effect with Word Clustering Features

Now that we showed the effect of using word-clustering features, we want to turn our attention to the shrinkage effect of these features on the models. Specifically, we measure the model size (second term in Eqn. 4) of each model that we presented in section 5.1. Thus, for each model, we measure the sum of absolute values of the predicted parameter weights and compare against different cluster sizes, and features sets.

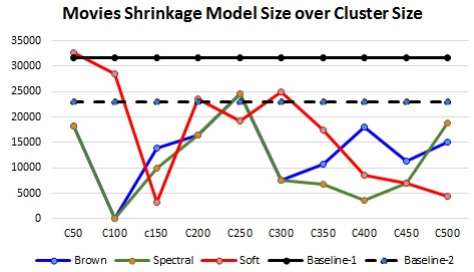

Figure 3: Model size, shrinkage, versus varying cluster sizes. Each line on each chart indicates the model sizes using different cluster sizes using baseline models as well as models with Brown, Spectral, and Soft clustering features.

In Figure 3, we show the model size results for different cluster size. Here we only show the results on the movies domain, although we observe the similar behavior on other domains as well. The results show the correlation between the shrinkage in the model size and performance improvement. Overall, the proposed features have smaller model sizes compared to the baseline models.

6. Conclusions

We applied a simple empirical rule for shrinking exponential models to the conditional random fields. We proposed several feature sets that shrink the model size to improve slot filling tasks. These feature sets consistently improved the slot filling accuracy across three tasks. It is well known that slot filling performance can be improved using additional information coming from dictionary, syntactic and/or semantic content of the sentence or various external resources. However typically they carry extra overhead either during training (e.g. manual labeling, dictionary acquisition) or decoding as those features should also be computed during decoding. There are several nice properties of the proposed feature sets: 1) automatically generated without any supervision, 2) almost no overhead during decoding (e.g. does not require running a parser to generate features), 3) task independence. We believe that the proposed features are also language independent and that they can achieve similar gains for slot filling tasks in other language. As part of our future work, we will apply the idea of shrinking model size along with the corresponding feature sets to other syntactic and semantic tagging tasks.

7. References

[1] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy, *Scaling Shrinkage-Based Language Models*, IEEE ASRU, December, 2009.

[2] J. Lafferty, A. McCallum, F. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, In Proc. ICML, pages 282-289, 2001.

[3] A. Deoras, R. Sarikaya, *Deep Belief Network based Semantic Taggers for Spoken Language Understanding*, In Proc. Interspeech, Lyon, France, 2013.

[4] P. Xu, R. Sarikaya, *Joint Intent Detection and Slot Filling with Convolutional Neural Networks*, In Proc. IEEE ASRU, Olomouc, Czech Republic, 2013.

[5] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai and R.L. Mercer, "Class-based N-gram Models of Natural Language", *Computational Linguistics*, 18(4), pp: 467–479, 1992.

[6] S. F. Chen, *Performance Prediction for Exponential Language Models*, In Proc. of HLT-NAACL, Boulder, CO, 2009.

[7] S. F. Chen, *Shrinking Exponential Language Models*, In Proc. of HLT-NAACL, Boulder, CO, 2009.

[8] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. "Inducing features of random fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:380393, 1997.

[9] L. Deng, G. Tur, X. He, and D. Hakkani-Tr, "Use of Kernel Deep Convex Networks and End-To-End Learning for Spoken Language Understanding", In Proc. of SLT 2012, IEEE Workshop on Spoken Language Technologies, Miami, FL, December 2012.

[10] S. F. Chen, and S. Chu, "Enhanced Word Classing for Model M", *Proc. of Interspeech*, Tokyo, Japan, 2010.

[11] R. Sarikaya, S. F. Chen, A. Sethy, B. Ramabhadran, "Impact of Word Classing on Shrinkage-based Language Models", *Proc. of Interspeech*, Tokyo, Japan, 2010.

[12] A. Emami, S. F. Chen, A. Ittycheriah, H. Soltau, B. Zhao, "Decoding with Shrinkage-based Language Models", *Proc. of Interspeech*, Tokyo, Japan, 2010.

[13] L. Bottou, " Large-Scale Machine Learning with Stochastic Gradient Descent", *Proc. of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, 177187, Edited by Yves Lechevallier and Gilbert Saporta, Paris, France, August 2010, Springer.

[14] K. Yao, J. Zweig, M. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding, *Proc. of Interspeech*, 2013.

[15] G. Tur, L. Deng, D. Hakkani-Tur, X. He, " Towards Deeper Understanding Deep Convex Networks for Semantic Utterance Classification," *Proc. of ICASSP*, Kyoto, Japan, 2012.

[16] R. Sarikaya, S. F. Chen, B. Ramabhadran, "Shrinkage Based Features for Natural Language Call Routing, *Proc. of Interspeech-2011* , Florence Italy, August 2011.

[17] A. Ng, et.al., "On Spectral Clustering: Analysis and An Algorithm", *Proc. of the NIPS*, 2001.

[18] J. Turian, L. Ratinov and Y. Bengio, "Word Representations: A simple and general method for semi-supervised learning", I*Proc. of ACL*, 2010.

[19] T. Anastasakos, Y-B. Kim, A. Deoras, "Task Specific Continuous Word Representations for Mono and Multi-lingual Spoken Language Understanding", *Proc. of ICASSP*, 2014.