Pre-training of Hidden-Unit CRFs

Young-Bum Kim[†] **Karl Stratos**[‡] **Ruhi Sarikaya**[†]

[†]Microsoft Corporation, Redmond, WA
[‡]Columbia University, New York, NY
{ybkim, ruhi.sarikaya}@microsoft.com
stratos@cs.columbia.edu

Abstract

In this paper, we apply the concept of pre-training to hidden-unit conditional random fields (HUCRFs) to enable learning on unlabeled data. We present a simple yet effective pre-training technique that learns to associate words with their clusters, which are obtained in an unsupervised manner. The learned parameters are then used to initialize the supervised learning process. We also propose a word clustering technique based on canonical correlation analysis (CCA) that is sensitive to multiple word senses, to further improve the accuracy within the proposed framework. We report consistent gains over standard conditional random fields (CRFs) and HUCRFs without pre-training in semantic tagging, named entity recognition (NER), and part-of-speech (POS) tagging tasks, which could indicate the task independent nature of the proposed technique.

1 Introduction

Despite the recent accuracy gains of the deep learning techniques for sequence tagging problems (Collobert and Weston, 2008; Collobert et al., 2011; Mohamed et al., 2010; Deoras et al., 2012; Xu and Sarikaya, 2013; Yao et al., 2013; Mesnil et al., 2013; Wang and Manning, 2013; Devlin et al., 2014), conditional random fields (CRFs) (Lafferty et al., 2001; Sutton and McCallum, 2006) still have been widely used in many research and production systems for the problems due to the effectiveness and simplicity of training, which does not involve task specific parameter tuning (Collins, 2002; McCallum and Li, 2003; Sha and Pereira, 2003; Turian et al., 2010; Kim and Snyder, 2012; Celikyilmaz et al., 2013; Sarikaya et al., 2014; Anastasakos et al., 2014; Kim et al., 2014; Kim et al., 2015a; Kim et al., 2015c; Kim et al., 2015b). The objective function for CRF training operates globally over sequence structures and can incorporate arbitrary features. Furthermore, this objective is convex and can be optimized relatively efficiently using dynamic programming.

Pre-training has been widely used in deep learning (Hinton et al., 2006) and is one of the distinguishing advantages of deep learning models. The best results obtained across a wide range of tasks involve unsupervised pre-training phase followed by the supervised training phase. The empirical results (Erhan et al., 2010) suggest that unsupervised pre-training has the regularization effect on the learning process and also results in a model parameter configuration that places the model near the basins of attraction of minima that support better generalization.

While pre-training became a standard steps in many deep learning model training recipes, it has not been applied to the family of CRFs. There were several reasons for that; (i) the shallow and linear nature of basic CRF model topology, which limits their expressiveness to the inner product between data and model parameters, and (ii) Lack of a training criterion and configuration to employ pre-training on unlabeled data in a task independent way.

Hidden-unit CRFs (HUCRFs) of Maaten et al. (2011) provide a deeper model topology and improve the expressive power of the CRFs but it does not address how to train them in a task independent way using unlabeled data. In this paper, we present an effective technique for pre-training of HUCRFs that can potentially lead to accuracy gains over HUCRF and basic linear chain CRF models. We cluster words in the text and treat clusters as pseudo-labels to train an HUCRF. Then we transfer the parameters corresponding to observations to initialize the training process on labeled

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 192–198,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

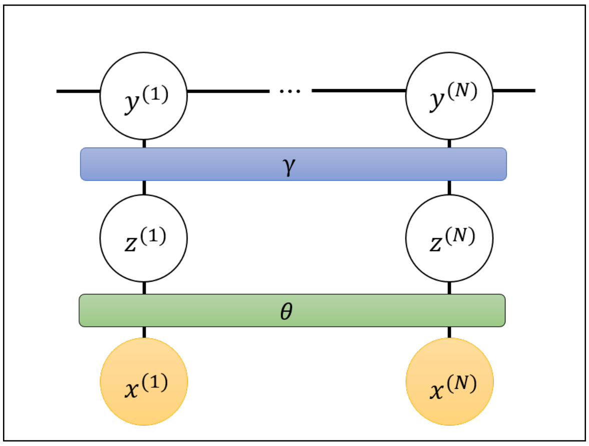

Figure 1: Graphical representation of hidden unit CRFs.

data. The intuition behind this is that words that are clustered together tend to assume the same labels. Therefore, learning the model parameters to assign the correct cluster ID to each word should accrue to assigning the correct task specific label during supervised learning.

This pre-training step significantly reduces the challenges in training a high-performance HUCRF by (i) acquiring a broad feature coverage from unlabeled data and thus improving the generalization of the model to unseen events, (ii) finding a good a initialization point for the model parameters, and (iii) regularizing the parameter learning by minimizing variance and introducing a bias towards configurations of the parameter space that are useful for unsupervised learning.

We also propose a word clustering technique based on canonical correlation analysis (CCA) that is sensitive to multiple word senses. For example, the resulting clusters can differentiate the instance of "bank" in the sense of financial institutions and the land alongside the river. This is an important point as different senses of a word are likely to have a different task specific tag. Putting them in different clusters would enable the HUCRF model to learn the distinction in terms of label assignment.

2 Model

2.1 HUCRF definition

A HUCRF incorporates a layer of binary-valued hidden units $z = z_1 \dots z_n \in \{0, 1\}$ for each pair of observation sequence $x = x_1 \dots x_n$ and label sequence $y = y_1 \dots y_n$. It is parameterized by

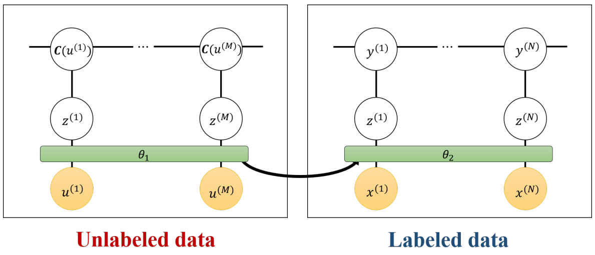

Figure 2: Illustration of a pre-training scheme for HUCRFs.

$\theta \in \mathbb{R}^d$ and $\gamma \in \mathbb{R}^{d'}$ and defines a joint probability of y and z conditioned on x as follows:

$$p_{\theta,\gamma}(y, z|x) = \frac{\exp(\theta^\top \Phi(x, z) + \gamma^\top \Psi(z, y))}{\sum_{\substack{z' \in \{0,1\}^n \\ y' \in \mathcal{Y}(x,z')}} \exp(\theta^\top \Phi(x, z') + \gamma^\top \Psi(z', y'))}$$

where $\mathcal{Y}(x, z)$ is the set of all possible label sequences for x and z, and $\Phi(x, z) \in \mathbb{R}^d$ and $\Psi(z, y) \in \mathbb{R}^{d'}$ are global feature functions that decompose into local feature functions: $\Phi(x, z) = \sum_{j=1}^{n} \phi(x, j, z_j)$ and $\Psi(z, y) = \sum_{j=1}^{n} \psi(z_j, y_{j-1}, y_j)$.

HUCRF forces the interaction between the observations and the labels at each position j to go through a latent variable z_j: see Figure 1 for illustration. Then the probability of labels y is given by marginalizing over the hidden units,

$$p_{\theta,\gamma}(y|x) = \sum_{z \in \{0,1\}^n} p_{\theta,\gamma}(y, z|x)$$

As in restricted Boltzmann machines (Larochelle and Bengio, 2008), hidden units are conditionally independent given observations and labels. This allows for efficient inference with HUCRFs despite their richness (see Maaten et al. (2011) for details). We use a perceptron-style algorithm of Maaten et al. (2011) for training HUCRFs.

2.2 Pre-training HUCRFs

How parameters are initialized for training is important for HUCRFs because the objective function is non-convex. Instead of random initialization, we use a simple and effective initialization scheme (in a similar spirit to the pre-training methods in neural networks) that can leverage a large body of unlabeled data. This scheme is a simple two-step approach.

In the first step, we cluster observed tokens in M unlabeled sequences and treat the clusters as labels to train an intermediate HUCRF. Let $C(u^{(i)})$ be the "cluster sequence" of the i-th unlabeled sequence $u^{(i)}$. We compute:

$$(\theta_1, \gamma_1) \approx \operatorname*{arg\,max}_{\theta,\gamma} \sum_{i=1}^{M} \log p_{\theta,\gamma}(C(u^{(i)})|u^{(i)}))$$

In the second step, we train a final model on the labeled data $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ using θ_1 as an initialization point:

$$(\theta_2, \gamma_2) \approx \operatorname*{arg\,max}_{\substack{\theta,\gamma: \\ \textbf{init}(\theta,\theta_1)}} \sum_{i=1}^{N} \log p_{\theta,\gamma}(y^{(i)}|x^{(i)})$$

While we can use γ_1 for initialization as well, we choose to only use θ_1 since the label space is task-specific. This process is illustrated in Figure 2.

In summary, the first step is used to find generic parameters between observations and hidden states; the second step is used to specialize the parameters to a particular task. Note that the first step also generates additional feature types absent in the labeled data which can be useful at test time.

3 Multi-Sense Clustering via CCA

The proposed pre-training method requires assigning a cluster to each word in unlabeled text. Since it learns to associate the words to their clusters, the quality of clusters becomes important. A straightforward approach would be to perform Brown clustering (Brown et al., 1992), which has been very effective in a variety of NLP tasks (Miller et al., 2004; Koo et al., 2008).

However, Brown clustering has some undesirable aspects for our purpose. First, it assigns a single cluster to each word type. Thus a word that can be used very differently depending on its context (e.g., "bank") is treated the same across the corpus. Second, the Brown model uses only unigram and bigram statistics; this can be an issue if we wish to capture semantics in larger contexts. Finally, the algorithm is rather slow in practice for large vocabulary size.

```
CCA-PROJ
Input: samples $(x^{(1)}, y^{(1)}) \dots (x^{(n)}, y^{(n)}) \in \{0,1\}^d \times \{0,1\}^{d'}$, dimension $k$
Output: projections $A \in \mathbb{R}^{d \times k}$ and $B \in \mathbb{R}^{d' \times k}$

  • Calculate $B \in \mathbb{R}^{d \times d'}$, $u \in \mathbb{R}^d$, and $v \in \mathbb{R}^{d'}$:
        $B_{i,j} = \sum_{l=1}^{n} [[x_i^{(l)} = 1]][[y_j^{(l)} = 1]]$
        $u_i = \sum_{l=1}^{n} [[x_i^{(l)} = 1]] \qquad v_i = \sum_{l=1}^{n} [[y_i^{(l)} = 1]]$
  • Define $\hat{\Omega} = \text{diag}(u)^{-1/2} B \text{diag}(v)^{-1/2}$.
  • Calculate rank-$k$ SVD $\hat{\Omega}$. Let $U \in \mathbb{R}^{d \times k}$ ($V \in \mathbb{R}^{d' \times k}$)
    be a matrix of the left (right) singular vector corresponding to the largest $k$ singular values.
  • Let $A = \text{diag}(u)^{-1/2} U$ and $B = \text{diag}(v)^{-1/2} V$.
```

Figure 3: Algorithm for deriving CCA projections from samples of two variables.

To mitigate these limitations, we propose multi-sense clustering via canonical correlation analysis (CCA). While there are previous work on inducing multi-sense representations (Reisinger and Mooney, 2010; Huang et al., 2012; Neelakantan et al., 2014), our proposed method is simpler and is shown to perform better in experiments.

3.1 Review of CCA

CCA is a general technique that operates on a pair of multi-dimensional variables. CCA finds k dimensions (k is a parameter to be specified) in which these variables are maximally correlated. Let $x^{(1)} \dots x^{(n)} \in \mathbb{R}^d$ and $y^{(1)} \dots y^{(n)} \in \mathbb{R}^{d'}$ be n samples of the two variables. For simplicity, assume that these variables have zero mean. Then CCA computes the following for $i = 1 \dots k$:

$$\operatorname*{arg\,max}_{\substack{a_i \in \mathbb{R}^d,\, b_i \in \mathbb{R}^{d'}: \\ a_i^\top a_{i'} = 0 \;\; \forall i' < i \\ b_i^\top b_{i'} = 0 \;\; \forall i' < i}} \frac{\sum_{l=1}^{n} (a_i^\top x^{(l)})(b_i^\top y^{(l)})}{\sqrt{\sum_{l=1}^{n} (a_i^\top x^{(l)})^2} \sqrt{\sum_{l=1}^{n} (b_i^\top y^{(l)})^2}}$$

In other words, each (a_i, b_i) is a pair of projection vectors such that the *correlation* between the projected variables $a_i^\top x^{(l)}$ and $b_i^\top y^{(l)}$ (now scalars) is maximized, under the constraint that this projection is *uncorrelated* with the previous $i - 1$ projections. A method based on singular value decomposition (SVD) provides an efficient and exact solution to this problem (Hotelling, 1936). The resulting solution $A \in \mathbb{R}^{d \times k}$ (whose i-th column is a_i) and $B \in \mathbb{R}^{d' \times k}$ (whose i-th column is b_i) can be used to project the variables from

Input: word-context pairs from a corpus of length n: $D = \{(w^{(l)}, c^{(l)})\}_{l=1}^{n}$, dimension k

Output: cluster $C(l) \leq k$ for $l = 1 \dots n$

- Use the algorithm in Figure 3 to compute projection matrices $(\Pi_W, \Pi_C) = \textbf{CCA-PROJ}(D, k)$.
- For each word type w, perform k-means clustering on $C_w = \{\Pi_C^\top c^{(l)} \in \mathbb{R}^k : w^{(l)} = w\}$ to partition occurrences of w in the corpus into at most k clusters.
- Label each word $w^{(l)}$ with the cluster obtained from the previous step. Let $\bar{D} = \{(\bar{w}^{(l)}, \bar{c}^{(l)})\}_{l=1}^{n}$ denote this new dataset.
- $(\Pi_{\bar{W}}, \Pi_{\bar{C}}) = \textbf{CCA-PROJ}(\bar{D}, k)$
- Perform k-means clustering on $\{\Pi_{\bar{W}}^\top \bar{w}^{(l)} \in \mathbb{R}^k\}$.
- Let $C(l)$ be the cluster corresponding to $Pi_{\bar{W}}^\top v^{(l)}$.

Figure 4: Algorithm for clustering of words in a corpus sensitive to multiple word senses.

the original d- and d'-dimensional spaces to a k-dimensional space:

$$x \in \mathbb{R}^d \longrightarrow A^\top x \in \mathbb{R}^k$$
$$y \in \mathbb{R}^{d'} \longrightarrow B^\top y \in \mathbb{R}^k$$

The new k-dimensional representation of each variable now contains information about the other variable. The value of k is usually selected to be much smaller than d or d', so the representation is typically also low-dimensional. The CCA algorithm is given in Figure 3: we assume that samples are 0-1 indicator vectors. In practice, calculating the CCA projections is fast since there are many efficient SVD implantations available. Also, CCA can incorporate arbitrary context definitions unlike the Brown algorithm.

3.2 Multi-sense clustering

CCA projections can be used to obtain vector representations for both words and contexts. If we wished for only single-sense clusters (akin to Brown clusters), we could simply perform k-means on word embeddings.

However, we can exploit context embeddings to infer word senses. For each word type, we create a set of context embeddings corresponding to all occurrences of that word type. Then we cluster these embeddings; we use an implementation of k-means which automatically determines the number of clusters upper bounded by k. The number of word senses, k, is set to be the number of label types occurring in labeled data (for each task-specific training set).

We use the resulting context clusters to determine the sense of each occurrence of that word type. For instance, an occurrence of "bank" might be labeled as "bank$_1$" near "financial" or "Chase" and "bank$_2$" near "shore" or "edge".

This step is for disambiguating word senses, but what we need for our pre-training method is the partition of words in the corpus. Thus we perform a second round of CCA on these disambiguated words to obtain corresponding word embeddings. As a final step, we perform k-means clustering on the disambiguated word embeddings to obtain the partition of words in the corpus. The algorithm is shown in Table 4.

4 Experiments

To validate the effectiveness of our pre-training method, we experiment on three sequence labeling tasks: semantic tagging, named entity recognition (NER), and part-of-speech (POS) tagging. We used L-BFGS for training CRFs [1] and the averaged perceptron for training HUCRFs. The number of hidden variables was set to 500.

4.1 Semantic tagging

The goal of semantic tagging is to assign the correct semantic tag to a words in a given utterance. We use a training set of 50-100k queries across domains and the test set of 5-10k queries. For pre-training, we collected 100-200k unlabeled text from search log data and performed a standard preprocessing step. We use n-gram features up to $n = 3$, regular expression features, domain specific lexicon features and Brown clusters. We present the results for various configurations in Table 1. HUCRF with random initialization from Gaussian distribution (HUCRF$_G$) boosts the average performance up to 90.52% (from 90.39% of CRF). HUCRF with pre-training with Brown clusters (HUCRF$_B$) and CCA-based clusters (HUCRF$_C$) further improves performance to 91.36% and 91.37%, respectively.

Finally, when we use multi-sense cluster (HUCRF$_{C+}$), we obtain an F1-score of 92.01%. We also compare other alternative pre-training methods. HUCRF with pre-training RBM

[1] For CRFs, we found that L-BFGS had higher performance than SGD and the average percetpron.

	alarm	calendar	comm.	note	ondevice	places	reminder	weather	home	avg
CRF	92.8	89.59	92.13	88.02	88.21	89.64	87.72	96.93	88.51	90.39
HUCRF$_{G}$	91.79	89.56	92.08	88.42	88.64	90.99	89.21	96.38	87.63	90.52
HUCRF$_{R}$	91.64	89.6	91.77	88.64	87.43	88.54	88.83	95.88	88.17	90.06
HUCRF$_{B}$	92.86	90.58	92.8	88.72	89.37	91.14	90.05	97.63	89.08	91.36
HUCRF$_{C}$	92.82	90.61	92.84	88.69	88.94	91.45	90.31	97.62	89.04	91.37
HUCRF$_{S}$	91.2	90.53	92.43	88.7	88.09	90.91	89.54	97.24	88.91	90.84
HUCRF$_{NS}$	90.8	89.88	91.54	87.83	88.15	91.02	88.2	96.77	89.02	90.36
HUCRF$_{C+}$	**92.86**	**91.94**	**93.72**	**89.18**	**89.97**	**93.22**	**91.51**	**97.95**	**89.66**	**92.22**

Table 1: Comparison of slot F1 scores on nine personal assistant domains. The numbers in boldface are the best performing method. Subscripts mean the following: G = random initialization from a Gaussian distribution with variance 10^{-4}, R = pre-training with Restricted Boltzmann Machine (RBM) using contrastive divergence of (Hinton, 2002), C = pre-training with CCA-based clusters, B = pre-training with Brown clusters, S = pre-training with skip-ngram multi-sense clusters with fixed cluster size 5, NS = pre-training with non-parametric skip-ngram multi-sense clusters, $C+$ = pre-training with CCA-based multi-sense clusters.

(HUCRF$_{R}$) does not perform better than with random initialization. The skip-gram clusters (HUCRF$_{S}$, HUCRF$_{SN}$) do not perform well either. Some examples of disambiguated word occurrences are shown below, demonstrating that the algorithm in Figure 3 yields intuitive clusters.

	NER		POS	
	Test-A	Test-B	Test-A	Test-B
CRF	90.75	86.37	95.51	94.99
HUCRF$_{G}$	89.99	86.72	95.14	95.08
HUCRF$_{R}$	90.12	86.43	95.42	94.14
HUCRF$_{B}$	90.27	87.24	95.55	95.33
HUCRF$_{C}$	90.9	86.89	95.67	95.23
HUCRF$_{S}$	90.18	86.84	95.48	95.07
HUCRF$_{NS}$	90.14	85.66	95.35	94.82
HUCRF$_{C+}$	**92.04**	**88.41**	**95.88**	**95.48**

Table 2: F1 Score for NER task and Accuracy for POS task.

word	**context**
Book	a **book(1)** store within 5 miles of my address find comic **book(1)** stores in novi michigan **book(2)** restaurant for tomorrow **book(2)** taxi to pizza hut look for **book(3)** chang dong tofu house in pocono find **book(3)** bindery seattle
High	restaurant nearby with **high(1)** ratings show me **high(1)** credit restaurant nearby the address for shelley **high(2)** school directions to leota junior **high(2)** school what's the distance to kilburn **high(3)** road domino's pizza in **high(3)** ridge missouri

Table 3: Examples of disambiguated word occurrences.

4.2 NER & POS tagging

We use CoNLL 2003 dataset for NER and POS with the standard train/dev/test split. For pre-training, we used the Reuters-RCV1 corpus. It contains 205 millions tokens with 1.6 million types. We follow same preprocessing steps as in semantic tagging. Also, we use the NER features used in Turian et al. (2010) and POS features used in Maaten et al. (2011).

We present the results for both tasks in Table 2. In both tasks, the HUCRF$_{C+}$ yields the best performance, achieving error reduction of 20% (Test-A) and 13% (Test-B) for NER as well as 15% (Test-A) and 8% (Test-B) for POS over HUCRF$_{R}$. Note that HUCRF does not always perform better than CRF when initialized randomly. However, However, HUCRF consistently outperforms CRF with the pre-training methods proposed in this work.

5 Conclusion

We presented an effective technique for pre-training HUCRFs. Our method transfers observation parameters trained on clustered text to initialize the training process. We also proposed a word clustering scheme based on CCA that is sensitive to multiple word senses. Using our pre-training method, we reported significant improvement over several baselines in three sequence labeling tasks.

References

Tasos Anastasakos, Young-Bum Kim, and Anoop Deoras. 2014. Task specific continuous word representations for mono and multi-lingual spoken language understanding. In *ICASSP*, pages 3246–3250. IEEE.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992.

Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Asli Celikyilmaz, Dilek Z Hakkani-Tür, Gökhan Tür, and Ruhi Sarikaya. 2013. Semi-supervised semantic tagging of conversational understanding using markov topic regression. In *ACL*, pages 914–923. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Anoop Deoras, Ruhi Sarikaya, Gökhan Tür, and Dilek Z Hakkani-Tür. 2012. Joint decoding for speech recognition and semantic tagging. In *INTERSPEECH*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *ACL*, volume 1, pages 1370–1380.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660.

Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Geoffrey Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *ACL*. Association for Computational Linguistics.

Young-Bum Kim and Benjamin Snyder. 2012. Universal grapheme-to-phoneme prediction over latin alphabets. In *EMNLP*, pages 332–343. Association for Computational Linguistics.

Young-Bum Kim, Heemoon Chae, Benjamin Snyder, and Yu-Seop Kim. 2014. Training a korean srl system with rich morphological features. In *ACL*, pages 637–642. Association for Computational Linguistics.

Young-Bum Kim, Minwoo Jeong, Karl Stratos, and Ruhi Sarikaya. 2015a. Weakly supervised slot tagging with partially labeled sequences from web search click logs. In *HLT-NAACL*, pages 84–92. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, Xiaohu Liu, and Ruhi Sarikaya. 2015b. Compact lexicon selection with spectral methods. In *ACL*. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015c. New transfer learning techniques for disparate label sets. In *ACL*. Association for Computational Linguistics.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Hugo Larochelle and Yoshua Bengio. 2008. Classification using discriminative restricted boltzmann machines. In *ICML*.

Laurens van der Maaten, Max Welling, and Lawrence K Saul. 2011. Hidden-unit conditional random fields. In *AISTAT*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL*, pages 188–191. Association for Computational Linguistics.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, pages 3771–3775.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *HLT-NAACL*, volume 4, pages 337–342. Citeseer.

Abdel-rahman Mohamed, Dong Yu, and Li Deng. 2010. Investigation of full-sequence training of deep belief networks for speech recognition. In *INTERSPEECH*, pages 2846–2849.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *EMNLP*. Association for Computational Linguistics.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Ruhi Sarikaya, Asli Celikyilmaz, Anoop Deoras, and Minwoo Jeong. 2014. Shrinkage based features for slot tagging with conditional random fields. In *Proc. of Interspeech*.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics.

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, pages 93–128.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Mengqiu Wang and Christopher D Manning. 2013. Effect of non-linear deep architecture in sequence labeling. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 78–83. IEEE.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for language understanding. In *INTERSPEECH*, pages 2524–2528.