Combining Labeled and Unlabeled Data with Word-Class Distribution Learning

Yanjun Qi
NEC Labs America Inc
4 Independence Way
Princeton, NJ 08540, USA.
yanjun@nec-labs.com

Ronan Collobert
NEC Labs America Inc
4 Independence Way
Princeton, NJ 08540, USA.
ronan@collobert.com

Pavel P. Kuksa
Computer Science Dept.
Rutgers University
Piscataway, NJ 08854, USA.
pkuksa@cs.rutgers.edu

Koray Kavukcuoglu
Computer Science Dept.
New York University
New York, NY 10003, USA.
koray@cs.nyu.edu

Jason Weston
NEC Labs America Inc
4 Independence Way
Princeton, NJ 08540, USA.
jaseweston@gmail.com

ABSTRACT

We describe a novel simple and highly scalable semi-supervised method called Word-Class Distribution Learning (WCDL), and apply it the task of information extraction (IE) by utilizing unlabeled sentences to improve supervised classification methods. WCDL iteratively builds class label distributions for each word in the dictionary by averaging predicted labels over all cases in the unlabeled corpus, and re-training a base classifier adding these distributions as word *features*. In contrast, traditional self-training or co-training methods add self-labeled *examples* (rather than features) which can degrade performance due to incestuous learning bias. WCDL exhibits robust behavior, and has no difficult parameters to tune. We applied our method on German and English name entity recognition (NER) tasks. WCDL shows improvements over self-training, multi-task semi-supervision or supervision alone, in particular yielding a state-of-the art 75.72 F1 score on the German NER task.

Categories and Subject Descriptors: I.2.7 [Artificial Intelligence]: Natural Language Processing - *text analysis*; M.4 [Knowledge Management]: Knowledge modeling

General Terms: Algorithms

Keywords: Semi-Supervised Learning, Semi-Supervised Feature Learning, Name Entity Recognition, Information Extraction

1. INTRODUCTION

Words are a fundamental building block in language and *features* based on words are a fundamental building block of natural language processing (NLP) systems. Indeed, many tasks such as named entity recognition (NER), part-of-speech (POS) tagging and chunking involve sequence modeling with word-level evaluation. For other tasks, such as document classification or sentiment extraction, that are evaluated at the document-level individual words still carry significant label information.

Supervised techniques using such features have yielded great success in the NLP community, but are restricted by the expense of annotating data. Popular semi-supervised methods such as self-training [18, 14, 12, 13] or co-training [2, 4] that utilize large unlabeled corpora try to improve over supervised methods by iteratively adding self-labeled *examples* predicted by the current model. However, they are vulnerable to the incestuous training bias problem [19, 21], i.e. examples may be consistently mislabeled making the model even worse on the next iteration. To combat this several authors have proposed schemes for only adding examples that meet a selection criterion [13, 19, 6], but these heuristic choices still might yield unreliable results.

In this paper we propose a novel semi-supervised strategy that works by providing semi-supervision at the level of *words* rather than *examples*. Under the assumption that words carry label information we measure the class label distribution for each word on a large unlabeled corpus. These features are then used to retrain the model in an iterative fashion. As noisy self-labeled examples are *not* added (as in self-training), our model exhibits robust behavior, and moreover has no difficult parameters (e.g. selection criteria) to tune.

We applied this strategy on two CoNLL-2003 [8] shared tasks, German NER and English NER. Using a state-of-the-art neural network model [5] in various setups, we observed improvements from using our method, called Word-Class Distribution Learning (WCDL), compared to the baseline classifier and to self-training, whenever we applied it. In particular we achieved a state-of-the art result of 75.72 F1 on the German NER task.

2. SEMI-SUPERVISED LEARNING WITH WORD-CLASS DISTRIBUTIONS

Unlike most popular semi-supervised approaches (details in Section 3), we propose to induce features from a large corpus of unannotated examples in a supervised fashion, and then use these features to augment the feature space of the labeled set.

2.1 Word-Class Distribution Learning

We consider the setting where one is given labeled training examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\ldots,L} \in \mathcal{X} \times \mathcal{Y}$ and an unlabeled set of examples $\{\mathbf{x}_i^*\}_{i=1,\ldots,U} \in \mathcal{X}$ where $U \gg L$. In particular $\mathcal{X}$ is the set of all sequences composed of elements which take on a finite

CIKM'09, November 2–6, 2009, Hong Kong, China.

set of possible values, e.g. sequences of words (but in the general case this could include other discrete types of feature as well, e.g. POS tags, stem-ends, etc.). That is, we will assume an input sequence $x = (x_1, \ldots, x_{|x|})$, where $x_j \in \mathcal{D}$, a dictionary of size $|\mathcal{D}|$. The labels $\mathcal{Y} \in \{1, \ldots, K\}$ are the K classes a sequence can be assigned to.

We define the word-class distribution for a given word $w \in \mathcal{D}$ as a vector $\text{wcd}(w) \in \mathbb{R}^K$ where

$$\text{wcd}(w)_i = P(y = i | w \in x). \quad (1)$$

That is, the i^{th} dimension measures the probability of label $y = i$ being assigned given that word w is present in the input sequence x. This distribution is of course unknown but can be estimated from the training set or, critically, can be *re-estimated* using unlabeled data by applying a trained classifier. We thus define the empirical word-class distribution as:

$$\overline{\text{wcd}}(w)_i = \frac{|\{j : f(x_j^*) = i \ \wedge \ w \in x_j^*\}|}{|\{k : w \in x_k^*\}|}, \quad (2)$$

where $f(\cdot)$ is a classifier trained to predict $y \in \mathcal{Y}$ given $x \in \mathcal{X}$.

We hence propose the following iterative semi-supervised training algorithm:

1. Define the feature representation $\phi(w)$ for a word w, and the representation $\Phi(x) = (\phi(x_1), \ldots, \phi(x_{|x|}))$ for an example x.
2. Train a classifier $f(\cdot)$ on training examples $(\mathbf{x}_i, \mathbf{y}_i)$ using the feature representation $\Phi(\cdot)$.
3. Augment the representation of words with their word-class distributions:

   $$\overline{\phi}(w) = (\phi(w), \overline{\text{wcd}}(w))$$

   using the current model $f(\cdot)$ to compute (2) and redefine $\Phi(x) = (\overline{\phi}(x_1), \ldots, \overline{\phi}(x_{|x|}))$.
4. Iterate steps 2 and 3.

2.2 Sequence Labeling with WCDL

In this work we consider sequence labeling tasks where inputs x are windows of a fixed size, and the middle word in the window is the word to be tagged. In this case one may want to consider the *modified word class distribution* where we are interested in class distributions only for the words to be labeled:

$$\overline{\text{wcd}}(w)_i = \frac{|\{j : f(x_j^*) = i \ \wedge \ w = (x_j^*)_m\}|}{|\{k : w = (x_k^*)_m\}|}, \quad (3)$$

where we only count matches to word w with the middle word with index $m = (|x_j^*| - 1)/2 + 1$. However, we still augment all words in the window with $\overline{\text{wcd}}(\cdot)$ features to capture local patterns.

2.3 Why Is It Useful?

Like self-training and co-training our algorithm (i) iteratively tries to improve its model; and (ii) is a wrapper approach that can use an supervised (or semi-supervised) classifier as a "base learner". However, our algorithm also has the following benefits:

- It has no incestuous bias from introducing new examples with incorrect labels as in self-training, as no examples are added.
- It does not require tricky selection heuristics as in self-training algorithms.
- The supervised model can *learn* if the $\overline{\text{wcd}}$ features are relevant or not (it can ignore or downweight them if it wants).
- The constructed $\overline{\text{wcd}}$ features contain information about the potential label of an example containing these words. This is collected by averaging over many unlabeled examples hence infrequent mistakes can be smoothed out and potentially corrected on the next iteration.
- In a sequence labeling task, the $\overline{\text{wcd}}$ features for neighboring words are highly informative for the word to be labeled.
- This algorithm is highly scalable (it adds a few features to the model, not lots of extra examples).

3. PREVIOUS WORK

We have already mentioned self-training [18] (also called 'bootstrapping' in the traditional NLP field) and co-training [2]. These methods augment the training set with labeled examples from the unlabeled set which are predicted by the model itself. This may give improvements in a model, but care must be taken as the predictions are prone to noise.

Many other semi-supervised learning algorithms exist, including tranductive SVMs [11], graph-based regularization [22], entropy regularization [10] and EM with generative mixture models [16], see [3] for a review. Apart from self-training and co-training, many other semi-supervised methods have scalability problems for realistic language modeling tasks, which normally involve hundreds of thousands of labeled examples.

Beyond the above approaches of semi-supervised learning with small amounts of labeled data and larger sets of unlabeled data, there has been a growing interest in the use of human-provided associations of features to particular classes for augmenting standard supervised learning. Most of this type of work has focused on using prior class-bias based features (called "labeled features") to generate labeled pseudo-examples or make feature selections [17, 20]. Further, the authors of [7] introduced a generalization expectation criterion to softly constrain the model's predictions on unlabeled examples with labeled features directly.

Finally, there are some methods that use auxiliary tasks on a large unlabeled corpus for training sequence models (often through multi-task learning). Ando et al. [1] proposed a method based on defining multiple tasks using unlabeled data that are multi-tasked with the task of interest, which they showed to perform very well on several natural language tagging tasks. Similarly, Collobert et al. [5] proposed a related method for deep neural networks where each word in the dictionary is represented by a vector (a representation which is shared between the multiple tasks). They multi-task with an unsupervised language model (LM), predicting the missing word in the middle of a text window, again resulting in good performance. In this work we follow the setup of [5] and measure the performance of WCDL as a wrapper on their approach.

4. EXPERIMENTS

We test our approach on the English and German NER datasets provided by the CoNLL-2003 shared task [8]. NER systems label atomic elements in the sentence into categories such as 'PERSON', 'COMPANY', or 'LOCATION', an important sub-task of information extraction (reviewed in [15]). For each language, a training set, a development set (for parameter tuning), a test set are provided. For both languages, more than 200,000 training tokens exist in the provided training sets (Table 1). The large unannotated ECI data file provided by CoNLL-2003 is used as our unlabeled corpus for the German NER. A sampled set of English Wikipedia web pages is used for WCDL on the English NER (size listed in Table 1).

Table 1: Number of (labeled) and unlabeled tokens used in our experiments in two CoNLL-2003 [8]) NER tasks .

Tokens Size in Task	Training (Labeled)	Unlabeled
German NER	206,931	∼58M
English NER	203,621	∼200M

4.1 Method

As a "base classifier" for the tagging task, we use the unified Neural Network (NN) framework of [5] where the input sentence is processed by several layers of feature extractions. The first layer maps words to 50-dimensional vectors (one vector for each word in the dictionary), the parameters of which are automatically *trained* during the learning process using backpropagation. The second layer is a classical layer of H hidden units (where H is optimized on the development set), and the final layer outputs probabilities of the class labels. In [5], the authors described its application to several well known NLP tasks including POS tagging and semantic role labeling, but do not report results for NER. They report using multi-tasking with an unsupervised task of learning a language model (LM) yields good results for other sequence labeling tasks. We hence tried using the LM as well.

We train our NER labeling system using a sliding window, optionally followed by a viterbi decoding of the entire sentence given the class probabilities from the NN predictions. This viterbi decoding could capture the local dependencies between targeted NER classes, which improves the NN performance effectively. The proposed WCDL (under sequence labeling) functions similarly as the viterbi decoding, since the learned class-distributions of surrounding words should obey local dependencies as well. Hence we compare WCDL with the viterbi step.

In all cases, it is straightforward to use WCDL. For all words in the text window centered at the target word, WCDL input features are concatenated along with the other word feature vectors.

Our baseline model uses the following word features:

- For English NER, we use (i) words in a 7-word-window surrounding the current word, (ii) capitalization flags of the current and surrounding words (Caps); and (iii) gazetteer information, as provided by CoNLL-2003;
- For German NER, we use (i) words in a 5-word-window surrounding the current word, (ii) capitalization flags of the current and surrounding words, (iii) prefix and suffix (length up to 4) of the current and surrounding words, (iv) the POS tags of the current and surrounding words; and (v) the chunk tag of the current and surrounding words. (Note in this case no gazetteer lists are used).

4.2 Results

We compare WCDL over multiple baselines, including NER by supervision alone, supervision with viterbi decoding, semi-supervision with LM, and with self-training.

Comparison with Supervision & Semi-supervision: Table 2 lists the test set performance on the German NER task using the F1 measure when applying WCDL as a wrapper to various systems: using only word features (with and without a viterbi decoding step), and using all features plus the language model (LM) based semi-supervised learning. In all cases WCDL improves over the baseline. Our best performance of 75.72 (using all features + WCDL) beats the state-of-the-art German NER performance of 75.27 which was reported in [1]. The best result during the CoNLL-2003 competition was 74.17 [9].

We also considered taking our best model, and adding the WCDL features predicted by it to a basic word-features only model. This improved its accuracy from 50.61 to 64.1. Using the LM as well yields 72.45 (words+LM on their own are 69.05). This is interesting because these results do not require POS, chunk, stem or caps features any more, but are close to the state-of-the-art.

Table 2: F1 score on the test set for German NER. For each choice of baseline (left column) applying word-class distribution learning (WCDL) improves over it (right column). LM means using language model semi-supervision.

Method	Baseline	+WCDL
Words only	45.89	**51.10**
Words only + Viterbi	50.61	**53.46**
All Features + LM	72.44	**73.32**
All Features + LM + Viterbi	74.33	**75.72**

Table 3: F1 score on the test set for English NER. WCDL improves over each baseline.

Method	Baseline	+WCDL
Words + Caps	77.82	**79.38**
Words + Caps + Viterbi	80.53	**81.51**
All Features + LM	86.49	**86.88**
All Features + LM + Viterbi	88.40	**88.69**

Table 3 provides results for the English NER task. Again, WCDL improves over all baselines; our best result was 88.69. In contrast, the best performing method during the competition was 88.76, and [1] have since reported 89.31 using multi-task semi-supervision. Here, our slightly worse performance seems to be due to our weaker baseline method (before even applying WCDL) compared to these approaches.

Comparison with Self-training We applied self-training to the same baseline methods to compare the performance of WCDL. There are numerous variants of self-training. We adopt the following weighting scheme: given L training examples, we choose L/R (R is a parameter to choose) unlabeled examples to add in the next round's training. By varying R, we get a range of impacts from self-training.

Table 4 and Table 5 give the results of the English and German NER. Self-training only helped marginally, or not at all, depending on the parameters.

Table 4: Test F1 of German NER using Self-Training.

Method	Baseline	R=1	R=10	R=100
Words only	**50.61**	47.07	47.92	47.9
All+LM	74.33	73.42	**74.41**	73.9

Table 5: Test F1 of English NER using Self-Training.

Method	Baseline	R=1	R=20	R=100
Words only	80.53	79.51	**81.01**	80.85
All+LM	**88.40**	87.64	88.07	88.17

The above comparison indicates that WCDL has better behavior than self-training with a random selection strategy. Since there exist many selection strategies for self-training, other selection techniques might bring improvements, see e.g. [6, 19] for other strategies. Still, these heuristic choices are difficult and need careful tuning [19]. In contrast, the proposed WCDL method does not seem to suffer from these issues.

Further, the performance in multiple rounds of self-training might oscillate because of degradation by noisy labels (see e.g. [19, 21]). We observed that WCDL's iterative training gives stable results. Figure 1 shows the test F1 from the iterations (as a wrapper for the "All features + LM + Viterbi" baseline) for the German NER set. It appears to converge in only a few iterations.

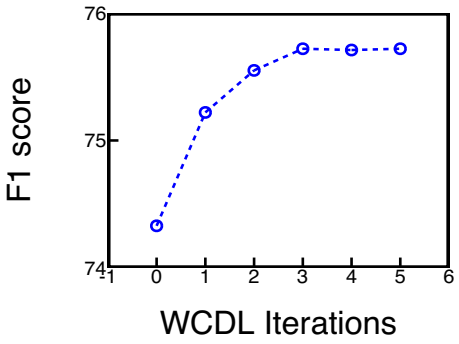

Figure 1: Test F1 over WCDL rounds on German NER.

5. CONCLUSIONS

In this work we proposed a novel semi-supervised algorithm called word class-distribution learning and applied it to the task of sequence labeling. Our method is highly scalable, contains no difficult parameters to tune, and we found it to be empirically robust, improving over every supervised and semi-supervised baseline method we applied it to.

The proposed method can easily be extended to other cases or domains. For example, instead of calculating predicted class distributions for each word, we could consider n-gram distributions instead. Moreover, one can generalize beyond word-level evaluation tasks. For instance in text categorization problems (document classification or sentiment analysis) a word's class distribution is the distribution of labels of *documents* that contained that word.

6. REFERENCES

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, New York, NY, USA, 1998. ACM.

[3] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. MIT Press, 2006.

[4] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on EMNLP*, pages 100–110, 1999.

[5] R. Collobert and J. Weston. A unified architecture for nlp: deep neural networks with multitask learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[6] H. Daumé III. Cross-task knowledge-constrained self training. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 680–688, Honolulu, Hawaii, October 2008. ACL.

[7] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602. ACM, 2008.

[8] Erik and F. De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147, 2003.

[9] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In W. Daelemans and M. Osborne, editors, *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 168–171, 2003.

[10] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 529–536, Cambridge, MA, 2005. MIT Press.

[11] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[12] R. J. Kate and R. J. Mooney. Semi-supervised learning for semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Short Papers (NAACL/HLT-2007)*, pages 81–84, 2007.

[13] Z. Kozareva, B. Bonev, and A. Montoyo. Self-training and co-training applied to spanish named entity recognition. In *MICAI 2005: Advances in Artificial Intelligence*, 2005.

[14] D. McClosky, E. Charniak, and M. Johnson. Effective self-training for parsing. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159, 2006.

[15] D. Nadeau and S. Sekine. A survey of named entity recognition and classfication. *Linguisticae Investigationes*, 30(1):3–26, 2007.

[16] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. volume 39, pages 103–134, Hingham, MA, USA, 2000. Kluwer Academic Publishers.

[17] R. E. Schapire, M. Rochery, M. G. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 538–545, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[18] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[19] H. Shan and D. Gildea. Self-training and co-training for semantic role labeling: Primary report. Technical Report TR891, University of Rochester, Comp. Sci. Dept., 2006.

[20] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 326–333, New York, NY, USA, 2004. ACM.

[21] T. Zhang, F. Damerau, and D. Johnson. Text chunking using regularized winnow. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 539–546, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[22] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03: Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003.