Keep your bearings: Lightly-supervised Information Extraction with Ladder Networks that avoids Semantic Drift

Ajay Nagesh
Dept. of Computer Science
University of Arizona
ajaynagesh@email.arizona.edu

Mihai Surdeanu
Dept. of Computer Science
University of Arizona
surdeanu@email.arizona.edu

Abstract

We propose a novel approach to semi-supervised learning for information extraction that uses ladder networks (Rasmus et al., 2015). In particular, we focus on the task of named entity classification, defined as identifying the correct label (e.g., person or organization name) of an entity mention in a given context. Our approach is simple, efficient and has the benefit of being robust to semantic drift, a dominant problem in most semi-supervised learning systems. We empirically demonstrate the superior performance of our system compared to the state-of-the-art on two standard datasets for named entity classification. We obtain between 62% and 200% improvement over the state-of-art baseline on these two datasets.

1 Introduction

Training machine learning systems with limited supervision is one of the fundamental challenges in natural language processing (NLP), as annotated data is often scarce and generating it requires costly human supervision. Semi-supervised learning addresses this challenge by combining limited supervision with a large, unannotated dataset, thereby mitigating the supervision cost.

For NLP, bootstrapping is a popular approach to semi-supervised learning due its relative simplicity coupled with reasonable performance (Abney, 2007). However, a crucial limitation of bootstrapping, which is typically iterative, is that, as learning advances, the task often *drifts semantically* into a related but different space, e.g., from learning women names into learning flower names (McIntosh, 2010; Yangarber, 2003).

In this paper, we propose an effective technique for semi-supervised learning for information extraction (IE), which obviates the need for an iterative approach, thereby mitigating the problem of semantic drift. Our technique is based on the recently proposed ladder networks (LNs) (Rasmus et al., 2015; Valpola, 2014). Ladder networks are deep denoising auto-encoders which have skip connections and reconstruction targets in the intermediate layers. Ladder networks are closely related to hierarchical latent variable models (Rasmus et al., 2015; Valpola, 2014). The lateral skip connections relieve the pressure on lower layers of the encoder to encode all latent information, thereby making the architecture modular in design, similar to a factor graph. The integration of the encoder-decoder framework as a neural network, allows one to use backpropagation for training, thereby not having to rely on intractable inference as in a standard graphical model. Furthermore, LNs have been shown to achieve state-of-the-art performance in image recognition tasks (Rasmus et al., 2015).

To the best of our knowledge, our work is one of the first applications of LN to any NLP task. Specifically, our contributions are as follows:

(1) We provide a novel application of LNs to an IE task, in particular semi-supervised named entity classification (NEC). Our approach is simple: we concatenate embeddings of entity mentions with that of its context[1] and feed the resulting vectors into the LN's denoising auto-encoder.

(2) We empirically demonstrate, for the task of semi-supervised NEC on two standard datasets – CoNLL (Tjong Kim Sang and De Meulder, 2003) and Ontonotes (Pradhan et al., 2013) – that we obtain a classification accuracy of 66.11% and 63.12% with minimal supervision on only 0.3% and 0.6% of the data, respectively. These results compare favorably against the accuracy of state-of-the-art bootstrapping algorithms of 40.74% and 21.06% on the same datasets. Further, in our experiments we observed an almost 7-fold decrease

[1] A context consists of all the patterns of n-grams within a certain window around the corresponding entity mention.

in training time compared to an iterative bootstrapping system.

(3) Lastly, we also provide empirical evidence that our approach is robust to the phenomenon of semantic drift. We obtain consistently better accuracy compared to traditional bootstrapping algorithms and label propagation, when initialized with identical supervision. We also demonstrate the reduction in semantic drift by measuring the purity of the entity pools with respect to a category as the algorithm advances (§4).

2 Related Work

There is a long line of work in semi-supervised learning for NLP (Zhu, 2005; Abney, 2007). This encompasses many different types of techniques such as self-training or bootstrapping (Carlson et al., 2010a,b; McIntosh, 2010; Gupta and Manning, 2015, inter alia), co-training (Blum and Mitchell, 1998), or graph-based methods such as label propagation (Delalleau et al., 2005). Perhaps the most popular approach among them is self-training, or bootstrapping, which has been used in many applications, including information extraction (Carlson et al., 2010a; Gupta and Manning, 2014, 2015), lexicon acquisition (Neelakantan and Collins, 2015), named entity classification (Collins and Singer, 1999) and sentiment analysis (Rao and Ravichandran, 2009). However, most of these approaches are iterative, and suffer from semantic drift (Komachi et al., 2008).

Auto-encoder frameworks have been getting a lot of attention in the machine learning community recently. Such framewoks include recursive auto-encoders (Socher et al., 2011), denoising auto-encoders (Vincent et al., 2008), etc. They are primarily used as a pre-training mechanism before supervised training. Recently, such networks have also been used for semi-supervised learning as they are more amenable to combining supervised and unsupervised components of the objective functions (Zhai and Zhang, 2015).

Ladder networks (LN) are stacked denoising auto-encoders with skip-connections in the intermediate layers (Rasmus et al., 2015; Valpola, 2014). LNs have been shown to produce state-of-the-art performance on both supervised and semi-supervised tasks on the MNIST dataset in image processing. Our work is among the first to apply LNs to NLP. While similar in spirit to Zhang et al. (2017), the only other work we found that applies a denoising auto-encoder to a semi-supervised spelling correction task, our work is much simpler, since it uses a multi-layer perceptron instead of convolution-deconvolution operations. Further, we demonstrate that LNs perform very well on a complex IE task, considerably outperforming several state-of-the-art approaches.

3 Approach

We apply the proposed semi-supervised learning approach to the task of NEC, defined as identifying the correct label of an entity mention in a given context. In our setting, the context of a mention is defined as all the patterns that match the specific mention. Please refer to the right half of Figure 1 for an example sentence snippet, an entity mention (in boldface) and its context. Using these as input, the classifier must infer that the mention's correct label is `person`.[2]

For the NEC task, the embedding of a mention and its context is concatenated to produce X which is input to the ladder network to predict a label y for the particular entity mention.

Initializing the network

We initialize the words in the entities and patterns around them with pre-trained word embeddings. To obtain a single embedding for an entity mention and its context we: (a) average word embeddings to obtain a single embedding for the entity mention and each of its patterns; and (b) average the resulting pattern embeddings to produce the embedding of the corresponding context. We then concatenate the mention's embedding and context embedding to be given as input to the ladder network. This process is depicted schematically in the right part of Figure 1.

Architecture of the ladder network

Ladder Network (Rasmus et al., 2015) is a neural network architecture designed to use unsupervised learning as a scaffolding for the supervised task. It is a denoising autoencoder (DAE) with noise introduced in every layer. It consists of two sets of encoders, a clean one and another corrupted with noise, and a decoder. In addition, there

[2] Note that the NEC task can be defined at mention level, as defined above, or at entity level, i.e., identify all labels that apply to all mentions of a given entity. (e.g., "Washington" = {`person`, `location`}. Here we focus on mention classification, although in some of our evaluations we revert to entity classification, to be able to compare against other approaches.

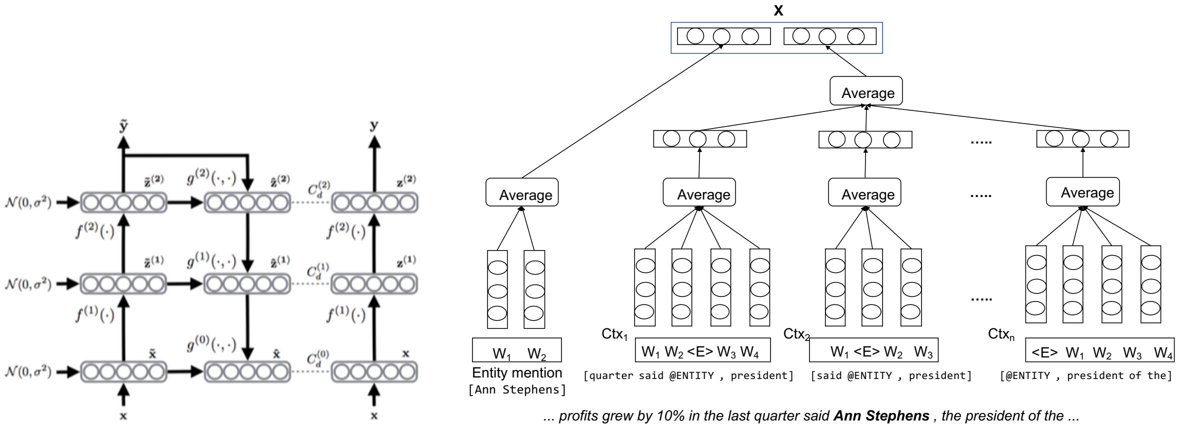

Figure 1: Architecture of the ladder network (Rasmus et al., 2015) (left) and of the network initialization component for the NEC task (right). LN is a deep denoising auto-encoder with lateral skip connections between the layers. The input to our LN is an entity mention along with its context, averaged and concatenated vector initialized with pre-trained embeddings for every token (§3). We introduce noise in the network by perturbing the embeddings with standard Gaussian noise with fixed stdev.

are skip connections between the encoder and decoder. The ladder network is defined as follows:

$$\tilde{X}, \tilde{Z}^{(1)}, \dots \tilde{Z}^{(L)}, \tilde{y} = f_{corr}(X) \tag{1}$$

$$X, Z^{(1)}, \dots Z^{(L)}, y = f_{clean}(X) \tag{2}$$

$$\hat{X}, \hat{Z}^{(1)}, \dots \hat{Z}^{(L)} = g(\tilde{Z}^{(1)}, \dots \tilde{Z}^{(L)}) \tag{3}$$

where X, $\tilde{X}$ and $\hat{X}$ is an input datapoint, its corrupted version, and its reconstruction, respectively; $Z^{(l)}$ and $\tilde{Z}^{(l)}$ are clean and corrupted hidden representations in the l-th layer; and, lastly, y, $\tilde{y}$ are the clean and corrupted activations, converted to a probability distribution over the label set (using a softmax layer). For our NEC task, X is the concatenation of an entity mention and its context embedding vectors generated as mentioned previously, and y represents one of the predicted mention labels (e.g. `person`).

We introduce noise in this architecture by perturbing the embeddings with a standard Gaussian noise with a fixed standard deviation.

The objective function is a combination of a supervised training cost and unsupervised reconstruction costs at each layer (including the hidden layers):

$$Cost = -\sum_{n=1}^{N} logP(\tilde{y_n} = y_n^* | X_n) + \sum_{n=N+1}^{M} \sum_{l=1}^{L} \lambda_l ReconstCost(Z_n^{(l)}, \tilde{Z}_n^{(l)}) \tag{4}$$

where the first term is the supervised cross-entropy based on the N labeled datapoints $(X_1, y_1^*), (X_2, y_2^*), \dots (X_N, y_n^*)$, and the second term is the reconstruction loss on the M unlabeled datapoints $X_{N+1}, X_{N+2}, \dots X_{N+M}$, for each layer l. Typically $M \gg N$.

Pezeshki et al. (2016) analyze the different architectural aspects of LN and note that the lateral connections and corresponding reconstruction costs (second term in Eq. 4) are critical for semi-supervised learning. In other words, it is important for unlabeled data to be used for regularization to be able to learn good abstractions in the different layers. We have similar observations for the NEC task (see Experiments). The overall architecture of LN is shown in the left part of Figure 1.

4 Experiments

Datasets: We used two datasets, the CoNLL-2003 shared task dataset (Tjong Kim Sang and De Meulder, 2003), which contains 4 entity types, and the OntoNotes dataset (Pradhan et al., 2013), which contains 11[3], both of which are benchmark datasets for supervised named entity recognition (NER). These datasets contain marked entity boundaries with labels for each marked entity. Here we only use the entity boundaries but *not* the labels of these entities during the training of our bootstrapping systems. To simulate learning from large texts, we tuned hyper parameters on development, but ran the actual experiments on the *train* partitions.

Baselines: We compared against 2 baselines:

Explicit Pattern-based Bootstrapping (EPB): this system is our implementation of the state-of-the-art bootstrapping system of Gupta and Manning (2015), adapted to NEC. The algorithm grows a pool of known entities and patterns for

[3] We excluded numerical categories such as `DATE`.

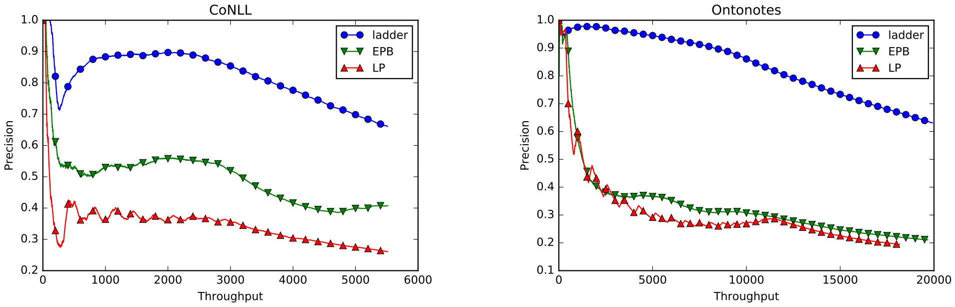

Figure 2: Overall results on the CoNLL (left) and Ontonotes (right) datasets. Throughput is the number of entities classified, and precision is the proportion of entities that were classified correctly.

each category of interest, from a few seed examples per category, by iterating between pattern promotion and entity promotion. The former is implemented using a ranking formula driven by the point-wise mutual information (PMI) between each pattern with the corresponding category; the top ranked patterns are promoted to the pattern pool in each iteration. The latter component promotes entities using a classifier that estimates the likelihood of an entity belonging to each class. Our feature set includes, for each category c: (a) edit distance between the candidate entity e and known entities for c; (b) the PMI (with c) of the patterns in the pool of c that matched e in the training documents; and (c) similarity between e and entities in c's pool in some semantic space.[4] Entities classified with the highest confidence for each class are promoted to the corresponding pool after each epoch.

Label Propagation (LP): we used the implementation available in the `scikit-learn` package of the LP algorithm (Zhu and Ghahramani, 2002).[5] In each bootstrapping epoch, we run LP, select the entities with the lowest entropy, and add them to their top category. Each entity is represented by a feature vector that contains the co-occurrence counts of the entity and each of the patterns that matches it in text.[6]

Settings: For each entity mention, we consider a n-gram window of size 4 on either side as a pattern. We initialized the mention and contexts embeddings input to the ladder network as well as the baseline system with pre-trained embeddings from Levy and Goldberg (2014) (size 300d) as this gave us improved results on the baseline compared to vanilla `word2vec` initialization. We used a 600d dimensional embedding for each datapoint (300 each from entity and context concatenated). We used a 3-layer ladder network with dimensions 600-500-K where K is the number of labels present in the dataset. Further, we used a standard Gaussian noise with stdev = 0.3 for the corrupted encoder and reconstruction cost for the 3-layers were 1000-10-0.1. We set the supervised examples (mentions along their corresponding contexts and labels) randomly. For CoNLL we used 40 and Ontonotes 440 examples, with equal representation from their labels' set. To compare with the baselines, which classify entities rather than mentions, we sorted the predictions returned by the LN in decreasing order of their activation scores and chose the most confident entity label (when all its mention scores were averaged). We ran the baselines until they predicted labels for all the entities. For the baselines, in each iteration we promoted 100 entities per category.[7] For a fair comparison, we used the same set of entity mentions as seeds (selected randomly) for each of our experiments.

Figure 2 shows the precision vs. throughput curves for the baselines and our LN approach. We see that on both the datasets the LN outperforms the baselines by a large margin. Further we notice that the LN is reasonably stable for most of the precision/recall curve whereas EPB degrades quickly. Iterative bootstrapping approaches inherently suffer from semantic drift: as the iterations progress the learned model begins to drift into a different semantic space due to incomplete statistics and ambiguity (McIntosh, 2010; Yangarber, 2003). These results parallel other previous observations that semantic drift is an inherent problem in iterative bootstrapping approaches (Komachi

[4] We used pre-trained word representations, averaged for multi-word entities, to compute cosine similarities between pairs of entities.

[5] `http://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelPropagation.html`

[6] We experimented with other feature values, e.g., pattern PMI scores, but all performed worse than raw counts.

[7] We also ran a cautious approach of promoting 10 entities per category per iteration and noticed that the former had better performance.

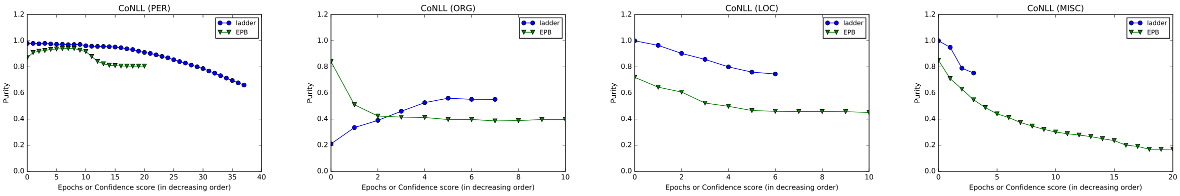

Figure 3: Avoiding semantic drift: Comparison of pool purity between ladder and EPB on the CoNLL dataset.

et al., 2008). The figure empirically demonstrates that, in contrast, the paradigm of semi-supervised learning based on ladder networks is more effective in combating semantic drift. Further, we empirically observed a speedup of almost 7x in training a ladder network compared to an iterative bootstrapping approach.

Table 1 lists the accuracy of the LN approach on all the data points, as we varied the amount of supervision. As expected, as we increase the amount of supervision, we observe improvements in accuracy. More importantly, the table shows that LN outperforms the overall accuracy of EPB (rightmost points in Figure 2) with much fewer annotations (e.g., with 55 annotations in OntoNotes, LN outperforms the performance of EPB with 440 annotated examples).

Figure 3 shows the purity of entity pools for a given label vs. confidence scores of the entity predictions sorted in decreasing order for the CoNLL dataset.[8] Purity is defined here as the precision of an entity pool for a given category. In the EPB setting, this is equivalent to computing the precision at the stage of entity promotion in a particular epoch. In LNs, we sort the entity predictions in decrease order of their confidence scores and create bins of size 100 for this comparison.We notice that for every category, LN maintains a higher overall purity over EPB, the best iterative bootstrapping baseline, demonstrating that the entity pools are less polluted by noisy entries, thereby reducing semantic drift. It is also important to observe that LN inherently captures the bias in the training data, by predicting more entities in the PER category, as this is the most frequently occurring label in the dataset.

5 Conclusion

We discussed a novel application of ladder networks to the task of lightly supervised named entity classification. Our approach concatenates embeddings of entity mentions with their contexts and feeds the resulting vectors into the LN's denoising auto-encoder. We demonstrate that our system outperforms state-of-the-art iterative bootstrapping approaches by approximately 62% and 200% on two benchmark datasets. Furthermore, our approach mitigates the issue of semantic drift as it is not iterative in nature, unlike traditional bootstrapping.

CoNLL		OntoNotes	
Num. labels	Accuracy	Num. labels	Accuracy
20	46.46	55	26.04
40	66.46	110	48.53
80	75.37	220	59.66
160	81.11	440	73.10
320	80.94	880	73.58
640	82.51	1760	73.23
1280	81.22	3520	73.77
2560	81.34	7040	73.31
5120	81.26	14080	82.47
10240	81.91	28160	83.32

Table 1: Num. of annotated labels vs. overall accuracy.
# of mention labels - CoNLL: 13200; OntoNotes: 67000

As part of future investigation, we will experiment with other types of encoders such as convolutional and recurrent networks. Furthermore, we aim to scale this approach to larger datasets. The approach presented in the paper is broad in scope. Application of this framework to other tasks in natural language processing such as relation extraction, sentiment analysis, and fine-grained entity typing, where obtaining supervised training data is hard, is another interesting avenue for further research. For example, relation extraction can be modeled similarly to the NEC task described here, as a feed forward network over embeddings of the entity mentions participating in the relation and of the lexico-syntactic patterns connecting them.

Acknowledgements

Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies. This work was funded by the DARPA Big Mechanism program under ARO contract W911NF-14-1-0395 and by the Bill and Melinda Gates Foundation HBGDki Initiative.

[8] In the appendix, a similar analysis is presented on the Ontonotes dataset.

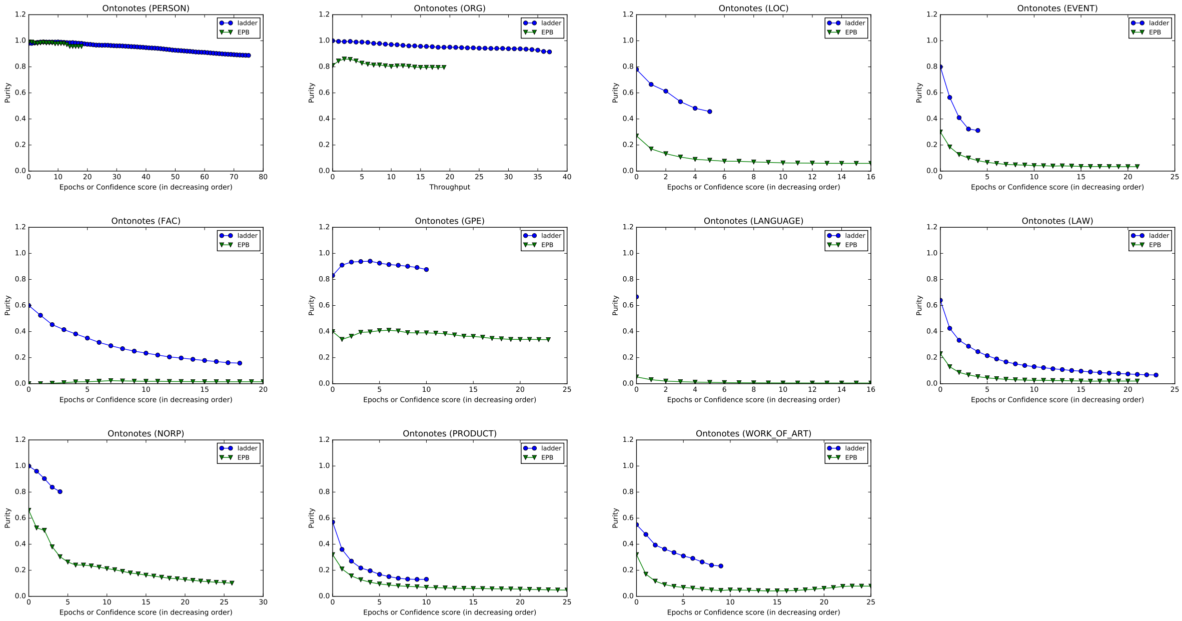

Figure 4: Avoiding semantic drift: Comparison of pool purity between ladder and EPB on the Ontonotes dataset.

References

Steven Abney. 2007. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st edition.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*. ACM, New York, NY, USA, COLT' 98, pages 92–100. https://doi.org/10.1145/279943.279962.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010a. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*.

Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010b. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, pages 101–110.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. 2005. Efficient non-parametric function induction in semi-supervised learning. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS'05)*. Society for Artificial Intelligence and Statistics, pages 96–103. http://www.iro.umontreal.ca/~lisa/pointeurs/semisup_aistats2005.pdf.

Sonal Gupta and Christopher D Manning. 2014. Improved pattern learning for bootstrapped entity extraction. In *CoNLL*. pages 98–108.

Sonal Gupta and Christopher D. Manning. 2015. Distributed representations of words to guide bootstrapped entity classifiers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pages 1011–1020. http://dl.acm.org/citation.cfm?id=1613715.1613847.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 302–308. http://www.aclweb.org/anthology/P14-2050.

Tara McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 356–365.

Arvind Neelakantan and Michael Collins. 2015. Learning dictionaries for named entity recognition using minimal supervision. *CoRR* abs/1504.06650. http://arxiv.org/abs/1504.06650.

Mohammad Pezeshki, Linxi Fan, Philemon Brakel, Aaron Courville, and Yoshua Bengio. 2016. Deconstructing the ladder network architecture. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, New York, New York, USA, volume 48 of *Proceedings of Machine Learning Research*, pages 2368–2376. http://proceedings.mlr.press/v48/pezeshki16.html.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Bjrkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Sofia, Bulgaria, pages 143–152. http://www.aclweb.org/anthology/W13-3516.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '09, pages 675–682. http://dl.acm.org/citation.cfm?id=1609067.1609142.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 3546–3554.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*. Edmonton, Canada, pages 142–147.

Harri Valpola. 2014. From neural PCA to deep unsupervised learning. *CoRR* abs/1411.7783. http://arxiv.org/abs/1411.7783.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '08, pages 1096–1103. https://doi.org/10.1145/1390156.1390294.

Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Shuangfei Zhai and Zhongfei Zhang. 2015. Semisupervised autoencoder for sentiment analysis. *CoRR* abs/1512.04466. http://arxiv.org/abs/1512.04466.

Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017. Deconvolutional paragraph representation learning. *CoRR* abs/1708.04729. http://arxiv.org/abs/1708.04729.

X. Zhu and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University. citeseer.ist.psu.edu/zhu02learning.html.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

A Purity on Ontonotes

Figure 4 shows the purity of the entity pools on the Ontonotes dataset (Purity is defined in §4). From these graphs, we can observe that LN has a higher overall purity compared to EPB for all categories, which indicates that it suffers less from the problem of semantic drift. Further, we observe that LN predicts more PERSON and ORG entities as these as the most frequently appearing types in this dataset. In other words, LN follows closely the underlying distribution of the data when making predictions, unlike EPB.