Domain Adaptation for Named Entity Recognition in Online Media with
Word Embeddings

Vivek Kulkarni* and Yashar Mehdad and Troy Chevalier
Yahoo Research
Sunnyvale, USA
{vvkulkarni@cs.stonybrook.edu, ymehdad@yahoo-inc.com, troyc@yahoo-inc.com}

Abstract

Content on the Internet is heterogeneous and arises from various domains like News, Entertainment, Finance and Technology. Understanding such content requires identifying named entities (persons, places and organizations) as one of the key steps. Traditionally Named Entity Recognition (NER) systems have been built using available annotated datasets (like CoNLL, MUC) and demonstrate excellent performance. However, these models fail to generalize onto other domains like Sports and Finance where conventions and language use can differ significantly. Furthermore, several domains do not have large amounts of annotated labeled data for training robust Named Entity Recognition models. A key step towards this challenge is to adapt models learned on domains where large amounts of annotated training data are available to domains with scarce annotated data.

In this paper, we propose methods to effectively adapt models learned on one domain onto other domains using distributed word representations. First we analyze the linguistic variation present across domains to identify key linguistic insights that can boost performance across domains. We propose methods to capture domain specific semantics of word usage in addition to global semantics. We then demonstrate how to effectively use such domain specific knowledge to learn NER models that outperform previous baselines in the domain adaptation setting.

This work was done when the author was a research intern at Yahoo.
© 2016 This is the author’s draft of the work. It is posted here for your personal use. Not for redistribution.

1 Introduction

Named Entity Recognition (NER) is a critical task for understanding textual content. While most NER systems demonstrate very good performance, this performance is typically measured on test data drawn from the same domain as the training data.

For example, most competitive Named Entity Recognition systems are trained on large amounts of labeled data from a given domain (like CoNLL or MUC) and evaluated on a held out test set drawn from the same domain [Florian et al., 2003; Chieu and Ng, 2002; Ando and Zhang, 2005; Collobert et al., 2011; Huang et al., 2015]. While such systems demonstrate high performance in-domain, content on the Internet can originate from multiple domains like Finance and Sports over which these systems perform quite poorly. Moreover one typically does not have access to large amounts of labeled examples on these domains to train robust domain specific models. This challenge is typically addressed through domain adaptation techniques (Blitzer et al., 2006; Jiang and Zhai, 2007; Satpal)
Most existing work on domain adaptation like Feature Sub-setting (Saptal and Sarawagi, 2007), Structural Correspondence Learning (Blitzer et al., 2006; Chen et al., 2012), learn a subset of features or learn dense representations of features that are more suited for domain adaptation. Different from these works, we explore word embeddings that explicitly capture domain specific differences while still capturing shared semantics across domains, and show that our proposed methods outperform several competitive baselines on domain adaptation for NER.

With recent advances in representation learning, word embeddings have been shown to be very useful features for several NLP tasks like POS Tagging, NER, and Sentiment Analysis (Chen et al., 2013; Al-Rfou et al., 2015; Collobert et al., 2011). One drawback of using generic word embeddings is that these word vectors do not capture domain specific differences in word semantics and usage. To illustrate this, consider articles from two distinct domains: (a) Sports and (b) Finance. The word tackle in the Sports domain is generally associated with moves in football and used as “A defensive tackle”. However in the domain of Finance, tackle is used to indicate problem solving as in “The company needs to tackle the rising costs immediately”.

Explicitly modeling such domain specific differences allows us to capture linguistic variation between domains that serve as distinctive features to boost performance of a machine learning model on NLP tasks. In this work we propose methods to effectively capture domain specific differences of language. We then apply our methods to analyze domain specific differences in word semantics. Finally, we demonstrate the effectiveness of using domain specific word embeddings for the task of Named Entity Recognition in the domain adaptation setting. Figure 1 shows the domain specific differences captured by our method across two domains (a) Sports and (b) Finance. Observe how the domain specific embeddings that our method learns can easily capture the distinct usages of a word (in this case as a Person or an Organization). As we will show in Section 4, such distinctive representations can improve performance of Named Entity Recognition in different domains outperforming competitive baselines.

In a nut shell, our contributions are as follows:

- **Linguistic Variation across Domains**: Given a word \( w \) how does its usage differ across different domains? We analyze variation in word usage (semantics) across different domains like Finance and Sports using distributed word representations (Section 2.1).

- **NER systems for Sports and Finance**: We propose methods to effectively use such domain specific knowledge captured by word embeddings towards the task of Named Entity Recognition. In particular we show how to build state of the art NER systems for domains with scarce amount of annotated training data by adapting NER models learned primarily on domains with large amounts of annotated training data (Section 2.2).

2 Methods

In this section we propose (a) Two methods to model domain specific word semantics in order to explicitly capture linguistic differences between domains and (b) Two methods that use domain specific word embeddings to learn robust Named Entity Recognition models for different domains using domain adaptation.

### 2.1 Domain Specific Linguistic Variation

**DOMAINDIST**

Given a corpus \( C \) with \( K \) domains and vocabulary \( \mathcal{V} \), we seek to learn a domain specific word embedding \( \phi_k : \mathcal{V} \mapsto \mathbb{R}^d \) using a neural language model where \( k \in \{1 \cdots K\} \). We apply the method discussed in (Bamman et al., 2014; Kulkarni et al., 2015b) to learn domain specific word embeddings. We briefly describe this approach below as pertaining to learning domain specific embeddings. For each word \( w \in \mathcal{V} \) the model learns (1) A global embedding \( \delta_{MAIN}(w) \) for the word ignoring all domain specific cues and (2) A differential embedding \( \delta_k(w) \) that encodes deviations from the global embedding for \( w \) specific to domain \( k \). The domain specific embedding \( \phi_k(w) \) is computed as: \( \phi_k(w) = \delta_{MAIN}(w) + \delta_k(w) \). The global word embeddings are randomly initialized, while the differential word embeddings are initialized to 0. We use the Skip-gram objective function with hierarchical soft-max to learn the global and the differential embeddings. We set the learning

---

1 In Section 4, we differentiate ourselves from Bamman et al., 2014 by outlining a probabilistic method that uses this model to disambiguate the domain given a phrase that outlines the usage of a word. w.
rate $\alpha = 0.025$, context window size $m$ to 10 and word embedding size $d$ to be 100. An example of the domain specific linguistic variation captured by DOMAINDIST is illustrated in Figure 1.

**DOMAINSense**

Here we outline yet another method to capture semantic variation in word usage across domains. We model the problem as follows:

- **Sense Specific Embeddings** We assume each word $w$ has potentially $S$ senses where we seek to learn not only an embedding for each sense of $w$ but also infer what these senses are from the corpus $C$.

- **Sense Proportions in Domains** The usage of $w$ in each domain $d$ can be characterized by a probability distribution $\pi_d(w)$ over the inferred senses of $w$.

To learn sense specific embeddings, we use the Adaptive Skip-gram model proposed by Bartunov et al., 2015 to automatically infer (a) the different senses a word $w$ exhibits (b) a probability distribution $\pi(w)$ over the different senses a word exhibits in the corpus and (c) an embedding for each sense of the word. Specifically, we combine the sub-corpora of different domains to form a single corpus $C$. We then learn sense specific embeddings for each word $w$ in $C$ using the Adaptive Skip-gram model. We set the number of dimensions $d$ of the embedding to 100, the maximum number of senses a word has $S = 5$ and restrict the vocabulary to only words that occur more than 100 times.

Finally, given a word $w$ we quantify the difference in the sense usage of $w$ between two domains $d_i$ and $d_j$ as follows:

1. Disambiguate each occurrence of $w$ in $d_i$ and $d_j$ using the method described by Bartunov et al., 2015. We can then estimate the sense distribution of word $w$ in domain $d_i$, $\pi_{d_i}(w)$ as $Pr(\pi_{d_i}(w) = s) = \frac{\#_{d_i}(\text{Sense}(w)=s)}{\#_{d_i}(w)}$ where $\#_{d_i}(X)$ represents the count of number of times $X$ is true in domain $d_i$.

2. We then compute the Jenson-Shannon Divergence (JSD) between the sense distributions of the word $w$ between the two domains $d_i$ and $d_j$ to quantify the difference in sense usage of $w$ between these domains.

Table 1 shows a small sample of words along with their inferred senses using this method. Figure 2 then depicts the domain specific difference in the sense usages of goal as computed by DOMAINSense.

While both DOMAINDIST and DOMAINSense explicitly capture domain specific differences in word semantics, they differ in their underlying models. DOMAINDIST captures domain specific word semantic/usage by directly learning domain specific word representations. DOMAINSense on the other hand infers different senses of a word and learns an embedding for each sense. Domain specific differences are then modeled by differences in sense usage of the word across domains. To illustrate this difference, consider the word goal. DOMAINDIST will capture the fact that goal is associated with match, winning in Sports and capture this sense of goal in the Sports Specific Embedding. DOMAINSense in contrast will infer that goal has two senses overall (see Table 1) and then capture that in Sports both these senses are used. Moreover, the sense related to score is used 70% of the time while the sense associated with objective is estimated to be used 30% of the time in Sports. Finally, we empirically evaluate the effectiveness of both DOMAINDIST and DOMAINSense for the task of NER (Section 4).

**Error Bounds on Estimated JS Divergence**

Note that for any given word $w$, the empirical probability estimate computed as $Pr(\pi_{d_i}(w) = s)$ is estimated from its usage in the sample corpus and is hence a random variable. Since these estimates of probabilities are further used to compute the JS Divergence between the sense distributions over two domains $(d_i, d_j)$ this estimate is also a random
variable and demonstrates variance. We now provide theoretical bounds on the standard deviation of the computed Jenson Shannon Divergence.

**Lemma 1.** Let $n_a(w)$ and $n_b(w)$ be the total number of occurrences of a word $w$ in domains $d_a$ and $d_b$ respectively. The standard deviation in the JS divergence of the sense distribution of $w$ across this domain pair is $O\left(\frac{1}{\sqrt{n_a(w)}} + \frac{1}{\sqrt{n_b(w)}}\right)$

**Proof.** We provide the proof in the supplemental material.

The bound above implies the following: (a) We can quantify the uncertainty in our estimates based on the frequency of the words in the corpus and interpret our results with greater confidence. Very rare words would have larger deviations. (b) Depending on an application’s sensitivity to error, we can estimate the appropriate sample complexity needed.

### 2.2 Domain Adaptation for NER

In the previous section, we described methods to capture domain specific linguistic variation in word semantics/usage by learning word embeddings that are domain specific. In this section, we outline how to learn NER models for the various domains using such word embeddings as features.

As in previous works, we treat NER as a sequence labeling problem. To train, we use CRF-suite (Okazaki, 2007) with L-BFGS algorithm. We use a BILOU label encoding scheme. The features we use are listed in Table 2. Our main features are tokens and word embeddings, within a small window of the target token. We investigate using different kinds of embeddings listed below:

- **Generic Word2vec Embeddings:** We learn generic Skipgram embeddings using English Wikipedia.

- **Domain/Sense Specific Word Embeddings:** We experiment by using the embeddings learned using DOMAINDIST and DOMAINS.

### DOMAINEmbNER

Here, we outline the supervised domain adaptation method that uses domain specific word embeddings to learn NER models that significantly outperform other baselines on NER task in the domain adaptation setting. In this setting, we are interested in a Named Entity Recognition system for domain $T$. However training data available for domain $T$ is scarce but we have access to a source domain $S$ for which we have large number of training examples. We would like to perform domain adaptation by learning a model using the large amount of training data in source domain $S$ and adapt it to work well on the target domain $T$. There exist a number of methods for the task of supervised domain adaptation (Jiang, 2008). We use a simple method for this task outlined below:

1. Combine the training data from $S$ and $T$. Note again that $|S| >> |T|$ in our setting.
2. Extract the features outlined for training the CRF model as outlined in Table 2. Note that we experiment with different kinds of word embeddings and baselines.
3. Learn a CRF model using this training data.
4. Evaluate the learned CRF model on the domain specific test data set and report the performance.

As we will show in Section 4, using domain specific word embeddings improves the performance of NER on these target domains significantly and outperforms previous baselines for this task.

### ACTIVEDOMAINEmbNER

In this section, we describe how we can learn a Named Entity Recognition system, assuming we
3 Datasets

In this section, we outline details of the datasets we consider for our experiments.

Our datasets can be classified into 2 categories (a) Unlabeled data for learning word embeddings and (b) Labeled data for the task of NER, each of which we describe below.

3.1 Unlabeled Data

We use the following unlabeled data sets for the purpose of learning word embeddings. We consider (a) all sentences of English Wikipedia (b) a random sample of 1 Million articles from Yahoo! Finance restricting our language to only English and (c) a random sample of 1 Million articles from Yahoo! Sports restricting our language to only English.

have no labeled training data in the target domain. We can however request for a small number of examples B to be labeled by annotators. In such a setting, one can actively choose the set of examples that need to be labeled which will be most useful to learn a good model. We propose a method to actively label examples for the purpose of domain adaptation which we describe succinctly in Algorithm 1. In Section 3.2 we show that by merely asking for an editorial to label 1500 sentences, we can achieve performance close to state of art in this setting.

3.2 Labeled Data

We also use labeled data sets for the task of learning NER models which we summarize in Table 2.

4 Experiments

Here, we briefly describe the results of our experiments on (a) Domain Specific Linguistic Variation and (b) Domain Adaptation for Named Entity Recognition.

4.1 Domain Specific Linguistic Variation

Table 3 shows some of the semantic differences in word usage captured by DOMAINDIST. Observe that the method is able to capture words like quote, overtime, hurdles that have alternative meanings (semantics) in a domain. For example, the word hurdles means challenges in Finance but a kind of athletic race in Sports. In addition to capturing words that differ in semantics, note that DOMAINDIST also uncovers differing semantic usages of entities as well, as depicted in Figure 1. In the domain of Finance, Anthem refers to a health insurance company but Anthem in Sports dominantly refers to a song like a team anthem. In Figure 3 a sample set of words detected by DOMAINSENSE are shown. Note once again, that we are able to capture domain specific differences between words (both entities and non-entities). Furthermore, DOMAINSENSE is able to quantify the proportion of each word sense usage in various domains. For example, the word tackle is used exclusively in Finance as a verb that means to solve, whereas in Sports tackle is dominantly used to refer to an American football move. Note that in Sports, the sense of tackle that means to solve is only used 30% of the time.

This ability to capture differing entity roles (like
Organizations and Persons) provides an insight into the effectiveness of domain specific embeddings for improved performance on Named Entity Recognition.

### 4.2 Domain Adaptation for Named Entity Recognition

In this section, we report the results of using our DOMAINDIST and DOMAINSENSE word embeddings for the task of Named Entity Recognition on Finance and Sports Domains in the domain adaptation setting as described in Section 2. We also outline the baseline methods we compare to below:

**Baseline methods**

Since our setting is the setting of domain adaptation for Named Entity Recognition, we consider several competitive baselines for this task:

- **CoNLL-only Model**: We consider a simple baseline where we train a NER model only using CoNLL data and generic Wikipedia Embeddings without any adaptation to the target domain.
- **Feature Subsetting**: This domain adaptation method tries to penalize features which demonstrate large divergence between source and target domains (Satpal and Sarawagi, 2007). It is worth noting that this models the task of NER as a classification problem and not a structured prediction problem.\(^4\)
- **Online-FLORS**: FLORS learns robust representations of each word based on distributional features and counts, to boost performance across domains and treats the tagging problem as a classification problem. We also use a random sample of 100K unlabeled sentences from each domain which FLORS uses to enrich the robustness of representations learned. We consider a scalable version of FLORS (Yin et al., 2016).
- **FEMA**: FEMA (Yang and Eisenstein, 2015) learns low dimensional embeddings of the features used in a CRF model by using a variant of the Skipgram Model (Mikolov et al., 2013). These features can be used to learn a model for sequence tagging. While they demonstrate their method on Part of Speech tagging, the method itself is general and can be applied to other tasks like Named Entity Recognition as well and provide a nice replacement for word embeddings as features in a CRF model. We learned 100 dimensional FEMA embeddings in our experiment.\(^5\)

**Results and Discussion**

Table 5 shows the performance of our methods and other baselines on Finance and Sports. First note that a CoNLL-only model without any domain adaptation results in poor performance. Domain Adaptation methods like Feature Subsetting and FLORS that model Named Entity Recognition as a classification problem, rather than a sequence prediction problem perform even worse. In contrast FEMA which learns dense representations of CRF features which can then be used to learn a more robust CRF model (that is more suited to domain adaptation) yields an significantly improved F1 score of 67.70 on Finance and 82.48 on Sports respectively. Empirically we observe that using DOMAINSENSE embeddings improves the performance over the ConLL only model, but does not perform as well on this task (especially in Finance). We hypothesize while decomposing a word into multiple fine-grained senses is useful to capture semantic variation, using fine-grained sense embeddings for every word results in a overly complex decision space when used for tasks like NER. Finally observe that DOMAINDIST which learns domain specific embeddings (without explicitly decomposing words into their senses) outperforms all these methods in both domains. This superior performance results from the ability to capture useful broad domain specific differences more effectively.

In Table 6 we evaluate the performance of using domain specific word embeddings against using just generic Wikipedia based word embeddings as a function of available training data \(\alpha\). First observe that on an average, using DOMAINDIST word embeddings improves the performance over using generic Wikipedia based embeddings. Observe that in general, as the amount of training data in the target domain increases, the advantage (gain) of using domain specific embeddings reduces. For example, when only 10% of training data is available for Finance, using domain specific word embeddings results in F1 Score gain of 1.21. However when 90% of training data is available in Finance we get a small but still significant boost of 0.42 in the F1 score on using domain specific word embeddings.

\(4\) We use the open source implementation provided at [https://github.com/yiyang-gt/feat2vec](https://github.com/yiyang-gt/feat2vec).\(5\) For brevity we present results here only DOMAINDIST embeddings as DOMAINSIT embeddings are the best performing embeddings.
We explain this by noting that as the proportion of training data in the target domain increases, the model is able to pick up on domain specific cues and fine-tune its decision boundary better without needing to rely too much on the domain specific cues captured by the domain specific word embeddings.

Table 7 shows the performance of ACTIVE-DOMAIN EMBNER as a function of number of sentences we sought to be actively labeled. Note that merely requiring 1500 sentences to be manually annotated, we are able to achieve close to state of
| α    | Finance (Wiki) | Sports (Wiki) |
|------|----------------|---------------|
| 0.1  | 62.50          | 80.18         |
| 0.2  | 66.61          | 83.61         |
| 0.3  | 68.35          | 84.42         |
| 0.4  | 70.61          | 85.49         |
| 0.5  | 71.55          | 86.30         |
| 0.6  | 71.71          | 87.08         |
| 0.7  | 72.42          | 87.07         |
| 0.8  | 73.33          | 88.12         |
| 0.9  | 74.02          | 88.00         |

Table 6: Performance of DOMAIN EMB NER using DOMAIN DIST Embeddings versus Wikipedia Embeddings on NER task against different proportions of training data in target domain.

| #(Sents) | Training Data % | Finance (F1) | Sports (F1) |
|----------|-----------------|--------------|-------------|
| 500      | 7.7             | 67.68        | 83.94       |
| 1000     | 15.5            | 69.68        | 85.53       |
| 1500     | 23.2            | 69.79        | 86.78       |
| 2000     | 31.0            | 70.53        | 87.14       |

Table 7: Performance of ACTIVE DOMAIN DIST NER on the target domains of Finance and Sports using DOMAIN DIST as a function of actively labeled sentences.

art F1 performance (69.79 on Finance and 86.78 on Sports respectively) outperforming competitive baselines.

5 Related Work

Related work can be organized into two areas: (a) Socio-variational linguistics and (b) Domain Adaptation.

Socio-variational linguistics Several works study how language varies according to geography and time (Eisenstein et al., 2010; Eisenstein et al., 2011; Bamman and others, 2014; Bamman et al., 2014; Kim et al., 2014; Kulkarni et al., 2015a; Kenter et al., 2015; Gonçalves and Sánchez, 2014; Kulkarni et al., 2015b; Cook et al., 2014; Freermann and Lapata, 2016; Hamilton et al., 2016). Different from these studies, our work seeks to identify semantic changes in word meaning (usage) across domains with a focus on improving performance on an NLP task like NER. The methods outlined in (Kulkarni et al., 2015b; Bamman et al., 2014) are most closely related to our work. While we directly build on methods outlined by them we differentiate ourselves from their work in two ways. First, we show that the model proposed in (Bamman et al., 2014) is not only useful for learning domain specific word embeddings but the model itself can be utilized for the task of domain disambiguation by using an often neglected set of model parameters (the output vectors of the model).

Second, we also present a method to quantify the semantic variation in word usage by explicitly modeling differences in the usage of different senses of words. While the methods outlined in (Kulkarni et al., 2015b; Bamman et al., 2014) capture domain specific differences, they do not explicitly model the fact that words have multiple senses and their usage in a domain is a mixture of different proportions over these senses which can be explicitly quantified. Finally we apply these methods to identify and analyze semantic variation in word usage across domains like Sports and Finance, highlight interesting examples of such variation prevalent across these domains with applications to Named Entity Recognition.

Domain Adaptation There is a long line of work on domain adaptation (Evgeniou and Pontil, 2004; Ando and Zhang, 2005; Blitzer et al., 2006; Jiang and Zhai, 2007; Satpal and Sarawagi, 2007; Daumé III, 2009; Chen et al., 2012; Schnabel and Schütze, 2014; Yang and Eisenstein, 2015). Most of these works can be classified based on the strategies they use as follows: (a) Instance Weighting Methods (Satpal and Sarawagi, 2007; Jiang and Zhai, 2007) (b) Regularization based methods (Evgeniou and Pontil, 2004; Daumé III, 2009) and (c) Representation Induction (Blitzer et al., 2006; Chen et al., 2012; Schnabel and Schütze, 2014; Yang and Eisenstein, 2015). Our method of learning domain specific word embeddings in an unsupervised manner can be placed into this final category. Finally an excellent survey of various domain adaptation algorithms for NLP is provided by (Jiang, 2008; Li, 2012).

6 Conclusions

In this paper, we proposed methods to detect and analyze semantic differences in word usage across multiple domains. Our methods explicitly capture domain specific cues by learning word embeddings from unlabeled text and scale well to large web scale data sets. Furthermore, we outline methods that leverage such domain specific linguistic variation and knowledge effectively to boost performance on NLP tasks like Named Entity Recognition on domains with scarce training data and requiring domain adaptation. Our methods not only out-perform previous competitive baselines but also require a very small number of manually annotated sentences in the target domain to achieve competitive performance. We believe our work sets...
the stage for new directions and further research into applications that effectively model linguistic variation across domains to improve the performance, applicability, and usability of NLP systems analyzing the diverse content on the Internet.

Acknowledgments

We thank Akshay Soni, Swayambhoo Jain, Aasish Pappu and Kapil Thadani for valuable insights and discussions.

References

[Al-Rfou et al.2015] Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot: Massive multilingual named entity recognition. In SDM.

[Ando and Zhang2005] Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research, 6(Nov):1817–1853.

[Bamman et al.2014] David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 828–834, June.

[Bamman and others2014] David Bamman et al. 2014. Gender identity and lexical variation in social media. Journal of Sociolinguistics.

[Bamman et al.2014] David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 828–834, June.

[Bartunov et al.2015] Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2015. Breaking sticks and ambiguities with adaptive skipgram. arXiv preprint arXiv:1502.07257.

[Blitzer et al.2006] John Blitzer, Ryan McDonald, et al. 2006. Domain adaptation with structural correspondence learning. In Proceedings of the 2006 conference on empirical methods in natural language processing, pages 120–128. Association for Computational Linguistics.

[Chen et al.2012] Minmin Chen, Zhixiang Xu, et al. 2012. Marginalized denoising autoencoders for domain adaptation. arXiv preprint arXiv:1206.4683.

[Chen et al.2013] Yanqing Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2013. The expressive power of word embeddings. arXiv preprint arXiv:1301.3226.

[Chieu and Ng2002] Hai Leong Chieu and Hwee Tou Ng. 2002. Named entity recognition: a maximum entropy approach using global information. In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics.
A Supplemental Material

A.1 Domain Disambiguation

In this section, we outline a method DOMAINDIST++ that builds on DOMAINDIST to infer the domain given a phrase highlighting a word’s usage is likely to belong to. Specifically, given a finite set of domains \( D \), a word \( w \) and a set of context words \( T \), we would like to infer the most likely domain \( d \) which reflects the given usage of \( w \). As an illustration, suppose we have two potential domains \( D = \{ \text{SPORTS}, \text{FINANCE} \} \) and given the usage of word \( \text{loss as loss of the dividend} \) we would like to infer that this usage is most likely from the \( \text{FINANCE} \) domain.

First note that DOMAINDIST models the probability of a word \( o \) given a context word \( c \) as follows:

\[
Pr(o|c, D = d) = \frac{exp(u_o^T v_c)}{\sum_{v \in V} exp(u_o^T v)}
\]  

(1)

where \( D \) is discrete random variable that represents the particular domain used. The vector \( u_o \) corresponds to the output vector for \( o \) while \( v_c \) corresponds to the input vector (word embedding) for word \( c \) in domain \( d \).

Given a word \( o \) and its context word \( c \) (where \( c \) is a context word in the usage of \( o \)), we seek to compute the probability \( Pr(D = d|o, c) \). We decompose this as follows:

\[
Pr(D = d|o, c) \propto Pr(o|c, D = d) \\
Pr(c|D = d) Pr(D = d)
\]  

(2)

The first term in Equation [2] is modeled by Equation [1] which can be efficiently computed in \( O(|V|) \) using hierarchical soft-max (Mikolov et al., 2013). The second term in Equation [2] is easily estimated by computing the relative frequency of \( c \) in the corpus specific to domain \( d \). The final term is just a prior on the domains (which can be computed by relative sizes of the domain specific corpora or set to uniform).

To conclude, given a word-context pair, we can estimate the domain that characterizes this word-usage by

\[
\hat{d} = \arg\max_d Pr(D = d|o, c).
\]  

(3)

While we discuss how to disambiguate the domain given word-context pair the above can be trivially extended when multiple context words are given. In such a case, we make an independence assumption: Each word-context pair is independent. This decomposes the joint into a product of probabilities for each word context-pair (each of which can be computed by Equation [1]).

In order to evaluate our method for domain disambiguation, we consider the following three competitive approaches:

- **Unigram Model** One simple method to disambiguate the domain reflecting the usage of a word \( w \) along with its context words \( T \), is to estimate the probability of this phrase under a unigram language model specific to the domain. Specifically we estimate the domain as follows:

\[
\hat{d} = \arg\max_d Pr(w|D = d) \prod\limits_{c \in T} Pr(c|D = d)
\]  

(4)
• **DistanceMean (DM)** We consider a simple nearest neighbor based method. For each domain $d$, we compute a score, the mean cosine similarity between the word $w$ and the context words $c \in T$ and choose the domain with the higher score. In summary, we estimate the domain by:

$$\hat{d} = \arg\max_d \text{Score}(d),$$

(5)

where $\text{Score}(d)$ is given by:

$$\text{Score}(d) = \frac{1}{|T|} \sum_{c \in T} \text{CosineSim}(v_w, v_c),$$

(6)

Here $v_w$ and $v_c$ are the word embeddings for $w$ and $c$ specific to the considered domain $d$.

• **Context Vector Mean (CVM)** We consider yet another nearest neighbor based method. For each domain $d$, we first compute the mean context embedding $\overline{v}_T$ by averaging the domain specific word embeddings for each context word $c$. We then compute a score $\text{Score}(d)$ which is the cosine similarity between the domain specific word embedding for $w$ and the mean context vector. Therefore, we estimate the domain as follows:

$$\hat{d} = \arg\max_d \text{Score}(d)$$

(7)

where $\text{Score}(d) = \text{CosineSim}(v_w, \overline{v}_T)$ and $\overline{v}_T = \frac{1}{|T|} \sum_{c \in T} v_c$

To highlight the differences between these methods and our method, we evaluate these methods on a small but insightful dataset of 20 phrases which we show in Table 8. We describe our observations and conclusions briefly below:

• The Unigram model fails when the context words are not distinctive of domains and occur with similar frequencies in both domains (as illustrated by the first eight examples in Table 8).

• **DOMAINDIST++** is superior to Unigram and is competitive with other nearest neighbor baselines. One drawback of the nearest neighbor methods is that they do not explicitly and interpret-ably capture strength of domain membership which **DOMAINDIST++** captures. To illustrate, consider disambiguating the usage of the word *on* in *we are focused on the heats*. The word *on* does not really have distinctive semantic usages between the two domains SPORTS and FINANCE. **DOMAINDIST++** will assign close to equal probabilities of membership of *on* to both these domains thus naturally capturing the graded membership, which is not possible using nearest neighbor methods (DM and CVM).

A.2 Error bounds on computation of JS Divergence for **DOMAINSENSE**

**Lemma 1 (Lemma [1])**. Let $n_a(w)$ and $n_b(w)$ be the total number of occurrences of a word $w$ in domains $d_a$ and $d_b$ respectively. The standard deviation in the JS divergence of the sense distribution of $w$ across this domain pair is $O\left(\frac{1}{\sqrt{n_a(w)}} + \frac{1}{\sqrt{n_b(w)}}\right)$

**Proof.** Assume that a word $w$ has $S$ senses where the probability distribution over the senses in domain $d_a$ is given by $p = (p_1, p_2, \ldots, p_S)$ and that in domain $d_b$ is given by $q = (q_1, q_2, \ldots, q_S)$.

The JS Divergence between the probability distributions $p$ and $q$ is given by:

$$JS(p, q) = \sum_{i} p_i \log p_i + \sum_{i} q_i \log q_i - \frac{1}{2} \sum_{i} \frac{p_i + q_i}{2} \log \left(\frac{p_i + q_i}{2}\right)$$

(8)

Now note that each of $p_i$ and $q_i$ are sample MLE estimates of multinomial distribution. The standard deviations of each of these sample MLE estimates denoted by $\sigma_{p_i}$ and $\sigma_{q_i}$ are as follows:

$$\sigma_{p_i} = \sqrt{\frac{p_i^*(1-p_i^*)}{n_a(w)}}$$

(9)

$$\sigma_{q_i} = \sqrt{\frac{q_i^*(1-q_i^*)}{n_b(w)}}$$

(10)

where $p_i^*$, $q_i^*$ are the true values of the particular probabilities. In order to quantify the standard deviation in the resulting computation of $JS(p, q)$ which we denote by $\sigma_{JS(p, q)}$, we now apply the rule for propagation of uncertainty[6] which yields:

---

[6]We ignore covariance terms as each parameter is estimated independently.
This is a very easy **race** and you will not need any help.

Other than that **race** like normal

This is a basic **race**

All the strategies from the first **race** apply here strongly consider purchasing the car for this race

The key to success in this **race** is ability

The basic skills to win this **race** require hard work

Any legal age sex **race** we are focused on the **heats**

expectations are set high for the **heats**

**serve** as powerful market players

**tackle** the opponent with force

**loss** cents per share

**tackle** record one solo

strong **track** record and low turnover

business was producing **loss**

disappointed but proud of the **race**

Talk to him to start the **race**

**water heaters** **heats** on demand

**tackle** defensive prospect

| Phrase                                                                 | Unigram | DM | CVM | DOMAINDIST++ | True Domain |
|-----------------------------------------------------------------------|---------|----|-----|--------------|-------------|
| This is a very easy **race** and you will not need any help           | X       | ✓  | ✓   | ✓            | Sports      |
| Other than that **race** like normal                                   | X       | ✓  | ✓   | ✓            | Sports      |
| This is a basic **race**                                               | X       | ✓  | ✓   | ✓            | Sports      |
| All the strategies from the first **race** apply here strongly consider purchasing the car for this race | X       | ✓  | ✓   | ✓            | Sports      |
| The key to success in this **race** is ability                        | X       | ✓  | ✓   | ✓            | Sports      |
| The basic skills to win this **race** require hard work               | X       | ✓  | ✓   | ✓            | Sports      |
| Any legal age sex **race** we are focused on the **heats**            | ✓       | ✓  | ✓   | ✓            | Finance     |
| expectations are set high for the **heats**                           | ✓       | ✓  | ✓   | ✓            | Sports      |
| **serve** as powerful market players                                   | ✓       | x  | ✓   | ✓            | Finance     |
| **tackle** the opponent with force                                     | ✓       | ✓  | x   | ✓            | Sports      |
| **loss** cents per share                                               | ✓       | ✓  | ✓   | ✓            | Finance     |
| **tackle** record one solo                                             | ✓       | ✓  | ✓   | ✓            | Sports      |
| strong **track** record and low turnover                               | ✓       | ✓  | ✓   | ✓            | Finance     |
| business was producing **loss**                                        | ✓       | ✓  | ✓   | ✓            | Finance     |
| disappointed but proud of the **race**                                | ✓       | ✓  | ✓   | ✓            | Sports      |
| Talk to him to start the **race**                                      | ✓       | ✓  | ✓   | ✓            | Sports      |
| **water heaters** **heats** on demand                                  | ✓       | ✓  | ✓   | ✓            | Finance     |
| **tackle** defensive prospect                                          | ✓       | ✓  | ✓   | ✓            | Sports      |

Table 8: Set of words (in bold) and their usages disambiguated by all the 4 methods for evaluation. ✓ indicates the method predicted the correct domain while X indicates the method predicted the incorrect domain. The **Unigram** model is easily mis-led by common words across both domains which are not domain distinctive.

\[
\sigma_{JS}(p,q) = \sqrt{\sum \left( \frac{\partial JS(p,q)}{\partial p_i} \right)^2 \sigma_p^2 + \sum \left( \frac{\partial JS(p,q)}{\partial q_i} \right)^2 \sigma_q^2}
\]

(11)

The coefficients \( \frac{\partial JS(p,q)}{\partial p_i} \) and \( \frac{\partial JS(p,q)}{\partial q_i} \) are called the sensitivity coefficients. Substituting the computations for \( \sigma_p \) and \( \sigma_q \) in Equation 11 completes the proof. This fact that the uncertainty in the JS Divergence is inversely proportional to the square root of the sample size enables us to get reasonably accurate estimates by choosing an appropriate sample size.