Enhancing Semantic Role Labeling for Tweets Using Self-Training

Xiaohua Liu [‡] [†], Kuan Li[*] [§], Ming Zhou [†], Zhongyang Xiong[§]
[‡]School of Computer Science and Technology
Harbin Institute of Technology, Harbin, 150001, China
[§]College of Computer Science
Chongqing University, Chongqing, 400030, China
[†]Microsoft Research Asia
Beijing, 100190, China
[†]*{xiaoliu, mingzhou}@microsoft.com*
[§]*sloweater@163.com,* [§]*zyxiong@cqu.edu.cn*

Abstract

Semantic Role Labeling (SRL) for tweets is a meaningful task that can benefit a wide range of applications such as fine-grained information extraction and retrieval from tweets. One main challenge of the task is the lack of annotated tweets, which is required to train a statistical model. We introduce self-training to SRL, leveraging abundant unlabeled tweets to alleviate its depending on annotated tweets. A novel strategy of tweet selection is presented, ensuring the chosen tweets are both correct and informative. More specifically, the correctness is estimated according to the labeling confidences and agreement of two Conditional Random Fields based labelers, which are trained on the randomly evenly spitted labeled data; while the informativeness is in proportion to the maximum distance between the tweet and the already selected tweets. We evaluate our method on a human annotated data set and show that bootstrapping improve a baseline by 3.4% F1.

Introduction

Twitter [1] has become an important fresh information source, and has inspired recent research, such as influential Twitter user detection (Kwak et al. 2010), fresh links mining (Dong et al. 2010) and breaking news extraction (Sankaranarayanan et al. 2009) from tweets. Semantic Role Labeling (SRL) for tweets, which takes a tweet as input and identifies arguments with their semantic roles for every predicate, develops this line of research, representing a critical step towards fine-grained information extraction (e.g., events and opinions) from tweets.

SRL has been well studied on formal corpus like news. However, all state-of-the-art SRL systems suffer a dramatic drop in performance when tested on a new genre of text (Punyakanok, Roth, and Yih 2008). Partially, Liu et al. (2010) report that the F1 score of a state-of-the-art system trained on news corpus drops sharply to as low as 43.3% on tweets. They credit this to the huge difference in written style between news and tweets.

Some researchers are thus motivated to build SRL systems dedicated to tweets (Liu et al. 2010), which requires a large volume of annotated tweets or rules. Manually annotating such a corpus or writing those rules is tedious and prohibitively unaffordable. Several solutions to alleviate this issue are proposed: Zadeh Kaljahi (2010) propose to adapt self-training for SRL by employing balancing and preselection methods; Huang et al. (2010) represent the text using latent-variable language models to build an open-domain SRL system; and most recently, Liu et al. (2010) use word alignment to transfer predicate-argument information from news sentences to tweets.

Following Zadeh Kaljahi (2010), we propose to use self-training to tackle this challenge. Initially, a small amount of manually labeled data are used as seeds to train a system; Then both informative and high confidently correctly labeled tweets are chosen to augment its training set, based on which the system are repeatedly retrained. A novel strategy of tweet selection is adopted, ensuring the chosen tweets are both correct and informative: The correctness is estimated according to the labeling confidences and agreement of two Conditional Random Fields (CRF) (Lafferty, McCallum, and Pereira 2001) based labelers, which are trained on the randomly evenly spitted labeled data, respectively; while the informativeness is in proportion to the maximum distance between the tweet and the already selected tweets.

There are several remarkable differences between ours and Zadeh Kaljahi (2010). Firstly, in contrast to their random or simplicity based selection, our selection strategy relies on two independent labelers. If both labelers output the same result high confidently for a tweet, this tweet is regarded correctly labeled. Secondly, while selecting tweet as new training data, we consider not only its correctness, but also its informativeness, which is evaluated based on its content similarity to the selected training tweets. Finally, we use linear CRF models, not the Maximum Entropy models, which prove less effective than the former.

Our work is also partially inspired by Wu et al. (2009). They propose a bootstrapping algorithm for named-entities recognition (NER) that selects from unlabeled target domain bridging instances, which are informative about the target domain and easy to be correctly labeled as well. However,

[*] This work has been done while the author was visiting Microsoft Research Asia.

[1] http://twitter.com/

their study is in the context of domain adaption and both the informativeness and the correctness depend on a model trained on the source domain, which is unavailable in our setting.

1,167 tweets are manually annotated for blind test. Experimental results on this data set show that our method boots the baseline by 3.4% F1. It is also demonstrated that combining informativeness and correctness boosts the performance.

Our contributions are summarized as follows.

1. We introduce a self-training method to the task of SRL for tweets, which considers both the informativeness and correctness while selecting a new labeled tweet to the training dataset. Experimental results show that our method improves the baseline by 3.4% F1 on a human annotated dataset .
2. We propose to train two independent models and use their labeling confidence and agreement on a tweet to estimate its correctness. And a tweet's content similarity to the training data set is used to evaluate its informativeness.

Related Work

Related work falls into three categories: SRL for normal text (e.g., news), SRL for tweets and semi-supervised learning for SRL.

SRL for Normal Text

Since its first introduction by Gildea and Jurafsky (2002), SRL has attracted increasing attention owing to its usefulness to other NLP tasks and applications, such as information extraction, question answering, and machine translation (Surdeanu et al. 2003). With the public availability of annotated corpora, e.g., the PropBank (Kingsbury and Palmer 2003), and the dedicated CoNLL shared tasks (Carreras and Màrquez 2005; Surdeanu et al. 2008), many data driven approaches have been developed, among which the pipelined approach is the standard practice, i.e., dividing the task into several successive components such as argument identification, argument classification, global inference, etc., and conquering them individually (Xue 2004; Koomen et al. 2005; Cohn and Blunsom 2005; Punyakanok, Roth, and Yih 2008; Toutanova, Haghighi, and Manning 2005; 2008) .

Exceptions exist. For example, Màrquez et al. (2005) sequentially label the words according to their positions relative to an argument (i.e., inside, outside, or at the beginning); Vickrey and Koller (2008) simplify the input sentence by hand-written and machine learnt rules before SRL; some other approaches resolve all the sub-tasks at the same time by integrating syntactic parsing and SRL into a single model (Musillo and Merlo 2006; Merlo and Musillo 2008), or by using Markov Logic Networks (MLN) (Richardson and Domingos 2006) as the learning framework (Meza-Ruiz and Riedel 2009).

All the above methods mainly aim at normal text, and based on some sort of annotated corpus exists; in contrast, our method focuses on SRL on tweets, for which no annotated data is available.

SRL for Tweets

Liu et al. (2010) introduce the task of SRL for tweets. They map predicate semantic structures from news sentences to news tweets (tweets that report news) to obtain training data, based on which a tweet specific system was trained. A linear CRF model is used to integrate conventional features such as lemma and part-of-speech. There are two substantial differences between this work and ours. Firstly, Liu et al. (2010) focus on only news tweets while ours extend their scope to general tweets. It is worth noting that, news tweets represent only a small portion of all tweets, e.g., 15.6% according to our investigation, and that these tweets are generally easier for SRL, as partially evidenced by one of our experiments in which the F1 score of their system drops from 66.0% on news tweets to 41.4% on general tweets. Secondly, their method relies on word alignment and leverages similar news sentences. In contrast, our method uses self-training to exploit abundant unlabeled tweets.

Semi-supervised Learning for SRL

Semi-supervised learning uses both labeled and un-labeled data. It fits the scenario where labeled data is scarce and hard to construct while unlabeled data is abundant and easy to access. Self-training (Yarowsky 1995) is a typical semi-supervised learning method. It iteratively adds the data that meets certain criteria to its training set, and use the augmented training set to re-train its model. This algorithm has been successfully applied to a serials of NLP tasks, such as Reference Resolution (Ng and Cardie 2003), POS tagging (Clark, Curran, and Osborne 2003), and parsing (McClosky, Charniak, and Johnson 2006), NER (Wu et al. 2009) , and more recently SRL (Zadeh Kaljahi 2010) , in which three selection strategies are explored, i.e., simplicity based selection, random selection and balanced selection. Following this line of research, we adopt self-training in our work. However, we use a different selection strategy that considers both the correctness based on labeling confidence and agreement, and informativeness evaluated according to content similarity. This selection strategy differentiates our method from existing ones.

Task Definition

Given a tweet as input, our task is to identify every predicate, and for every predicate further identify its arguments. We use the general role schema defined by PropBank, which includes core roles such as A0, A1 (usually indicating the agent and patient of the predicate, respectively), and auxiliary roles such as AM-TMP and AM-LOC (representing the temporal and location information of the predicate, respectively). In our work, we only consider the verbal predicate, which is consistent with most existing SRL systems. Following Màrquez et al. (2005), we conduct word level labeling. As a pilot study, we focus on English tweets, though our method can straightforwardly extended to support tweets of other languages.

Our Method

Now we present our self-training method for the task of SRL for tweets. An overview of our method is first given, followed by detailed discussion of its core components.

Method Overview

Algorithm 1 outlines our method, where: $train$ denotes a machine learning process to get two independent statistical models l and $l^{'}$, both of which use linear CRF models [2]; the $label$ function generates predicate-argument structures with the help of the trained mode; p, s and cf denote a predicate, a set of argument and role pairs related to the predicate and the predicted confidence, respectively; the $select$ function tests if a labeled tweet meets the selection criteria; N and M are the maximum allowable number of new labeled training tweets and training data, respectively, which are experimentally set to 200 and 10,000, separately; the $shrink$ function keeps removing the oldest tweets from the training data set, until its size is less than M.

Algorithm 1 Self-training based SRL for tweets.

```
Require: Tweet stream $i$;training tweets $ts$;output stream $o$.
Initialize two CRF based labelers $l$ and $l^{'}$: $(l, l^{'}) = train(cl)$.
Initialize the number of new accumulated tweets for training $n$: $n = 0$.
while Pop a tweet $t$ from $i$ and $t \neq null$ do
    Label $t$ with $l$:$(t, \{(p, s, cf)\}) = label(l, c, t)$.
    Label $t$ with $l^{'}$:$(t, \{(p, s, cf)\}^{'}) = label(l^{'}, c, t)$.
    Output labeled results $(t, \{(p, s, cf)\})$ to $o$.
    if $select(t, \{(p, s, cf)\}, \{(p, s, cf)\}^{'})$ then
        Add $t$ to training set $ts$:$ts = ts \cup \{t, \{(p, s, cf)\}\}$;$n = n + 1$.
    end if
    if $n > N$ then
        Retrain labelers:$(l, l^{'}) = train(cl)$;$n = 0$.
    end if
    if $|ts| > M$ then
        shrink the training set:$ts = shrink(ts)$.
    end if
end while
```

Model

We choose linear CRF as our model with the following considerations: 1) Compared with classification models, it can jointly label multiple arguments including the word and its role, for a given predicate; 2) it has achieved the state-of-the-art results on the PropBank corpora (Màrquez et al. 2005); and 3) it is faster compared with its alternatives, such as MLN, which explores a far larger search space.

Following Màrquez et al. (2005), we use the BIO labeling schema. B, I, and O indicate the beginning, middle and out of an argument, respectively. Here is an example of a labeled sequence with this schema: "...<B-A0>earthquake<O> shorten<B-A1>day...". The above label sequence can be straightforwardly translated into predicate argument triples: $\{$(shorten,earthquake,A0),(shorten,day,A1)$\}$.

In our experiments, the CRF++[3] toolkit is used to train and test our linear CRF model.

Features

Before feature extraction, tweet meta data is extracted and normalized as well so that every link and account name become *LINK* and *ACCOUNT*, respectively. Hash tags are treated as common words. We then use conventional features defined in Màrquez et al. (2005), such as the lemma/POS tag of the current/previous/next token, the lemma of the predicate and its combination with the lemma/POS tag of the current token, the voice of the predicate (active/passive), the distance between the current token and the predicate, and the relative position of the current token to the predicate. Unlike Liu et al. (2010), dependencies parsing related features are used as well. The OpenNLP toolkit and the Stanford parser [4] are used to extract these features.

Selection Criteria

The selection procedure, as illustrated in Algorithm 2, consists of three steps. Firstly, the labeling results of two models are compared, and if they are not the same the tweet will not be selected, otherwise go to the next step for further checking. Secondly, check if the labeling confidence of any model is less than a threshold α (0.05 in our work). If yes, the tweet again will not be chosen; otherwise move forward to the next step. Finally check if the similarity between the tweet and any tweet in the training set is more than a threshold β (0.85 in our work). If yes, the tweet will not be considered; otherwise the tweet will finally be selected.

To compute the similarity between two tweets, both are first represented as bag-of-words vectors, in which stop words are removed. Stop words are mainly from a list of common words [5]. Then the cosine function is applied. Other similarity functions (e.g., Euclidean distance) and alternative weighting schema (e.g., weighting more on nouns and named entities) are tried, but no significant improvements are observed.

Experiments

In this section, we first introduce how the experimental data is prepared, and then evaluate our SRL system on the test data set. It will be demonstrated that our system outperforms the baseline, and that clustering boosts the performance.

Data Preparation

We use the Twitter API to crawl all tweets from April 20th 2010 to April 25th 2010, then drop non-English tweets and get about 11,371,389 tweets, from which 8, 000 tweets are randomly sampled. The selected tweets are then labeled by

[2] The labeled tweets are evenly and randomly divided into two parts, to train l and $l^{'}$, respectively.

[3] http://crfpp.sourceforge.net/

[4] http://nlp.stanford.edu/software/lex-parser.shtml

[5] http://www.textfixer.com/resources/common-english-words.txt

Algorithm 2 Selection of a training tweet.

```
Require: Training tweets $ts$; tweet $t$; labeled results by $l$ $\{(p,s,cf)\}$; labeled results by $l^{'}$ $\{(p,s,cf)\}^{'}$.
if $\{(p,s,cf)\} \neq \{(p,s,cf)\}^{'}$ then
    return FALSE.
end if
if $\exists cf \in \{(p,s,cf)\} \cup \{(p,s,cf)\}^{'} < \alpha$ then
    return FALSE.
end if
if $\exists t^{'} \in ts \; sim(t,t^{'}) > \beta$ then
    return FALSE.
end if
return TRUE.
```

two independent annotators following the annotation guidelines for PropBank, with one exception: For phrasal arguments, only the head word is labeled as the argument, to be consistent with the word level labeling system. 829 tweets are dropped because of inconsistent annotation, and finally 7,171 tweets are kept, forming the gold-standard data set. The gold-standard data set is randomly split into three parts: the first part consisting of 583 tweets is used as seeds for training, the second part with 5,421 tweets is used for self-training development, and the third part is used for blind test.

Evaluation Metrics

Following the common practice in SRL system evaluation, we adopt Precision (Pre.), recall (Rec.) and F1 as the evaluation metrics. Precision is a measure of what percentage the outputted labels are correct, and recall tells us to what percentage the labels in the gold-standard data set are correctly labeled, while F1 is the harmonic mean of precision and recall.

Performance of Reference Systems

Two off-the-shelf systems are studied to understand the domain mismatch problem, which motivates this work. One is the MLN based system (Meza-Ruiz and Riedel 2009), which is trained on the CoNLL08 shared task data set and achieves state-of-the-art performance on that task; the other is the first tweet specific system from Liu et al. (Liu et al. 2010), which is based on CRF as well, but focuses on news tweets and is trained on mechanically labeled tweets. The same toolkits (OpenNLP and the Stanford parser) is used to extract conventional features for the reference systems. Table 1 shows the performance of these two systems and ours on the same test data set, where SRL_{MLN}, SRL_{TN} and SRL_{SE} denote the MLN based system (Meza-Ruiz and Riedel 2009), the system from Liu et al. (Liu et al. 2010) and ours, respectively. Note that all these systems conduct word level SRL. From Table 1, it can been that ours performs remarkably better than SRL_{MLN} and SRL_{TN}. This is understandable since SRL_{MLN} is trained on formal text and SRL_{TN} on mechanically labeled news tweets.

We also use the test data set from Liu et al. (2010) to evaluate our method. The F1 is 67.1%, slightly better than that of SRL_{TN} (66.0%). This can be largely explained by the fact that our method, though not trained on news tweets, uses human labeled tweets and self-training. In future, we plan to self-learning to Liu et al. (2010), to see if it helps when only automatically labeled training data is available.

Table 1: Comparison to reference systems.

System	Pre.(%)	Rec.(%)	F1(%)
SRL_{SE}	59.2	45.9	51.7
SRL_{TN}	51.1	34.8	41.4
SRL_{MLN}	38.6	47.5	42.6

Table 2: Basic experimental results.

System	Pre.(%)	Rec.(%)	F1(%)
SRL_{SE}	59.2	45.9	51.7
SRL_{BA}	46.7	50.0	48.3

Baseline

A modified version of our system without self-training, hereafter denoted by SRL_{BA}, is adopted as the baseline. Compared with first tweet specific system from Liu et al. (Liu et al. 2010), this baseline is trained on human annotated data set, and use features related to dependency parsing.

Results

Table 2 shows the experimental results for the baseline and ours. From Table 2, it can be seen that clustering significantly boosts the F1 from 48.3% to 51.7% (with $p < 0.05$), suggesting the effectiveness of self-training. Table 3 presents detailed results of our method for different roles.

Effects of Self-training

It turns out that finally 2,557 in the development data set are selected for training, of which 544, or 21.3%, are completely correctly labeled. This largely explains the performance difference in Table 2. To reveal more details about the effectiveness of self-training, we feed development tweets into our method sequentially, and record the F1 score on the blind test data set immediately after every retraining, showing in Figure 1.

Effects of Correctness

The selection strategy is modified to ignore the correctness related conditions. Table 4 shows the corresponding results

Table 3: Experimental results of our method for different roles.

Role	Pre.(%)	Rec.(%)	F1(%)
A0	70.0	72.4	71.2
A1	52.0	47.5	49.6
A2	46.0	27.4	34.3
AM-TMP	40.4	12.2	18.8
AM-LOC	40.9	22.7	29.2
AM-NEG	87.5	63.6	73.7
OTHER	53.5	20.5	29.7

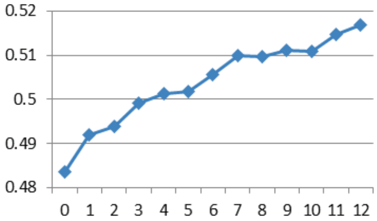

Figure 1: F1 score on the blind test dataset. Horizontal and vertical axes represent the number of retraining and the F1 score (%), respectively.

Table 4: Performance of two systems with and without considering correctness while selecting training tweet, respectively.

System	Pre.(%)	Rec.(%)	F1(%)
SRL_{SE}	59.2	45.9	51.7
SRL_{SE-C}	48.4	36.5	41.6

(denoted by SRL_{SE-C}), which are worse. It is found that the modified system selects more tweets for training from the development dataset, about 4,371, of which however only 9.8% are completely correctly labeled, largely explaining the performance difference.

Effects of Informativeness

Similarly, the selection strategy is modified to bypass the informativeness related conditions. Table 5 shows the results, where the second, and third column denotes the number of tweets selected for training and F1 score, respectively. It can be seen that with informativeness considered, our method give comparable results to the modified (denoted by SRL_{SE-I}), which however uses more training tweets and requires more training time.

Comparison with Other Selection Strategies

Other three selection strategies are explored: random selection, which randomly decides if a tweet should be selected; simplicity-based selection, which prefers simple tweets; and conventional confidence based selection. In our experiments, the selected probability is 0.2, which yields the best performance for random selection based systems; the simplicity of a tweet is estimated by the number of words in a tweet with stop-words and meta data removed, and tweets with less than 8 words are selected; the confidence threshold is experimentally set to 0.4. Table 6 show the results, where SRL_{RD}, SRL_{SP} and SRL_{CF} refer to the system with random, simplicity-based and confidence-based selection, respectively. From Table 6, it can be seen that out selection strategy is better than its alternatives.

Table 5: : Performance of two systems with and without considering informativeness while selecting training tweet, respectively.

System	#T	F1(%)
SRL_{SE}	2,557	51.7
SRL_{SE-I}	2,780	51.8

Table 6: Performance of systems with different selection strategies.

System	#T	F1(%)
SRL_{SE}	2,557	51.7
SRL_{RD}	2,557	47.7
SRL_{SP}	4,000	42.5
SRL_{CF}	4,277	44.9

Table 7: Performance of our method with varied minimum labeling confidence α.

α	0	0.02	0.05	0.1	0.15	0.2
F1(%)	51.1	51.6	51.7	51.6	51.3	51.4

Influence of Systematic Parameters

Table 7 and 8 show the performance of our method with varied α, the minimum labeling confidence, and β , the maximum similarity between the tweet and training tweets, separately, indicating that our method is not much sensitive to α or β. Table 9 shows the performance of our method with varied N, the maximum allowable number of the new accumulated training tweets. It can be seen that, $N = 200$ yields the best performance, but the performance difference caused by N is small.

Error Analysis

Errors made by our system can roughly be divided into three categories. The first kind of error, which constitutes 53.6% of all errors, is largely caused by the noisy features extracted or the irregular words in tweets. For example, for tweet "@JosieHenley thank youuuu sweedie pops !! Xxx", the POS tagger labels "@JosieHenley thank youuuu sweedie pops" as Proper Noun, Preposition, Pronoun, Verb, Noun, respectively, because of no punctuation following "@JosieHenley" and the irregular word "sweedie". These POS errors cause our system to ignore (thank, youuuu, A1) and to incorrectly recognize (sweedie, youuuu, A0) and (sweedie, pops, A1). Another example is the tweet "...im gonna arrest the mexicans...", in which the "'" between "i" and "m" is lost. Therefore, it is impossible for our system to correctly identify "i" as the A0 argument of "arrest". Developing tweet normalization technologies is a promising solution for this kind of error.

The second kind of error, which accounts for 31.5% of all errors, is mainly caused by data sparseness. For example, our system cannot correctly label this tweet "Bacteria in the

Table 8: Performance of our method with varied similarity threshold β.

β	0.75	0.80	0.85	0.90	0.95	1.00
F1(%)	51.2	51.2	51.7	51.5	51.5	51.8

Table 9: : Performance of our method with varied maximum allowable number of new accumulated training tweets.

N	**100**	**150**	**200**	**250**	**300**
F1(%)	51.5	50.9	51.7	51.6	51.4

gut shown to lower obesity : http://dld.bz/bDy", partially for the reason that the word "Bacteria" does not appear in our training data. Continually labeling more training data can help to fix these errors.

The third kind of error, which represents 14.9% of all errors, partially owes to the complexity of syntactic structures. For example, for this tweet "What are some famous quotes that you live by, or have changed http://url4.eu/2XUYp", our method incorrectly labels "http://url4.eu/2XUYp" as the A1 argument of "changed", because current features tell nothing about the existence of the subordinate clause in this tweet. Alleviating these kinds of errors requires syntactic features, which are not always available for all tweets because many tweets are grammatically incorrect. And it is inefficient and unnecessary to extract syntactic features for every tweet, since many tweets have simple syntactic structure, for which our system works pretty well. We are building a classifier to single out tweets with complex syntactic structures, and developing a specific SRL system for such tweets.

Conclusions and Future work

The task of SRL for tweets requires a large number annotated tweets, which is unavailable. Manually labeling such a corpus is however tedious and prohibitively unaffordable. To alleviate this issue, we introduce a self-training to this task. Our method selects potentially informative and correctly labeled tweets to enhance the training set, which is used to repeatedly retrain the SR labelers. In contrast to existing methods, the correctness is measured by both labeling confidence and agreement of two independently trained labelers while the informativeness is estimated according to the similarity between the tweet and the already selected training tweets. Experimental results show that our method improves the baselines by 3.4% F1.

In future, we plan to explore three directions to improve this work: 1) Adapting current POS systems to tweets; and 2) normalization technologies to clean tweets before SRL; 3)other selection conditions such as syntax complexity based, utility based selection and cross-validation based selection.

References

Carreras, X., and Màrquez, L. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *CoNLL*, 152–164.

Clark, S.; Curran, J. R.; and Osborne, M. 2003. Bootstrapping pos taggers using unlabelled data. In *HLT-NAACL*, 49–55.

Cohn, T., and Blunsom, P. 2005. Semantic role labelling with tree conditional random fields. In *CONLL*, 169–172.

Dong, A.; Zhang, R.; Kolari, P.; Bai, J.; Diaz, F.; Chang, Y.; Zheng, Z.; and Zha, H. 2010. Time is of the essence: improving recency ranking using twitter data. In *WWW*, 331–340.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Comput. Linguist.* 28:245–288.

Huang, F., and Yates, A. 2010. Open-domain semantic role labeling by modeling word spans. In *ACL*, 968–978.

Kingsbury, P., and Palmer, M. 2003. Propbank: The next level of treebank. In *Proceedings of Treebanks and Lexical Theories*.

Koomen, P.; Punyakanok, V.; Roth, D.; and Yih, W.-t. 2005. Generalized inference with multiple semantic role labeling systems. In *CONLL*, 181–184.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is twitter, a social network or a news media? In *WWW*, 591–600.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 282–289.

Liu, X.; Li, K.; Han, B.; Zhou, M.; Jiang, L.; Xiong, Z.; and Huang, C. 2010. Semantic role labeling for news tweets. In *Coling*, 698–706.

Màrquez, L.; Comas, P.; Giménez, J.; and Català, N. 2005. Semantic role labeling as sequential tagging. In *CONLL*, 193–196.

McClosky, D.; Charniak, E.; and Johnson, M. 2006. Effective self-training for parsing. In *NAACL*, 152–159.

Merlo, P., and Musillo, G. 2008. Semantic parsing for high-precision semantic role labelling. In *CoNLL*, 1–8.

Meza-Ruiz, I., and Riedel, S. 2009. Jointly identifying predicates, arguments and senses using markov logic. In *NAACL*, 155–163.

Musillo, G., and Merlo, P. 2006. Accurate parsing of the proposition bank. In *NAACL*, 101–104.

Ng, V., and Cardie, C. 2003. Weakly supervised natural language learning without redundant views. In *NAACL*, 94–101.

Punyakanok, V.; Roth, D.; and Yih, W.-t. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Comput. Linguist.* 34:257–287.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Mach. Learn.* 62:107–136.

Sankaranarayanan, J.; Samet, H.; Teitler, B. E.; Lieberman, M. D.; and Sperling, J. 2009. Twitterstand: news in tweets. In *GIS*, 42–51.

Surdeanu, M.; Harabagiu, S.; Williams, J.; and Aarseth, P. 2003. Using predicate-argument structures for information extraction. In *ACL*, 8–15.

Surdeanu, M.; Johansson, R.; Meyers, A.; Màrquez, L.; and Nivre, J. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL*, 159–177.

Toutanova, K.; Haghighi, A.; and Manning, C. D. 2005. Joint learning improves semantic role labeling. In *ACL*, 589–596.

Toutanova, K.; Haghighi, A.; and Manning, C. D. 2008. A global joint model for semantic role labeling. *Comput. Linguist.* 34:161–191.

Vickrey, D., and Koller, D. 2008. Applying sentence simplification to the conll-2008 shared task. In *CoNLL*, 268–272.

Wu, D.; Lee, W. S.; Ye, N.; and Chieu, H. L. 2009. Domain adaptive bootstrapping for named entity recognition. In *EMNLP*, 1523–1532.

Xue, N. 2004. Calibrating features for semantic role labeling. In *EMNLP*, 88–94.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 189–196.

Zadeh Kaljahi, R. S. 2010. Adapting self-training for semantic role labeling. In *ACL*, 91–96.