Collective Semantic Role Labeling for Tweets with Clustering

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

As tweets have become a comprehensive repository of fresh information, Semantic Role Labeling (SRL) for tweets has aroused great research interests because of its central role in a wide range of tweet related studies such as fine-grained information extraction, sentiment analysis and summarization. However, the fact that a tweet is often too short and informal to provide sufficient information poses a major challenge. To tackle this challenge, we propose a new method to collectively label similar tweets. The underlying idea is to exploit similar tweets to make up for the lack of information in a tweet. Specifically, similar tweets are first grouped together by clustering. Then for each cluster a two-stage labeling is conducted: One labeler conducts SRL to get statistical information, such as the predicate/argument/role triples that occur frequently, from its highly confidently labeled results; then in the second stage, another labeler performs SRL with such statistical information to refine the results. Experimental results on a human annotated dataset show that our approach remarkably improves SRL by 3.1% F1.

1 Introduction

Twitter [1] has become an important fresh information source, and has inspired a surge of studies on tweets, such as fresh links mining [Dong *et al.*, 2010], breaking news extraction [Sankaranarayanan *et al.*, 2009], and Semantic Role Labeling (SRL) [Liu *et al.*, 2010b]. The task of SRL for tweets usually takes a tweet as input and identifies arguments with their semantic roles for each predicate. This task is meaningful since it represents a critical step towards fine-grained information extraction (e.g., events and opinions), sentiment analysis and summarization for tweets.

SRL for tweets is a challenging task, largely owing to the fact that a tweet, less than 140 characters and freely published by anyone without proofreading, is often short and informal. This means that, NLP tools used to extract conventional features are not reliable when applied to tweets. For example, OpenNLP [2], the state-of-the-art part-of-speech (POS) tagger, achieves an accuracy of only 74.0% on tweets. This is also partially evidenced by the observation of Liu et al. [2010b] that the F1 score of a state-of-the-art system trained on a news corpus drops to 43.3% on tweets, as opposed to 90.0% on news. However, we observe that, though a tweet is often not informative, many similar (or almost identical) tweets tend to exist. For example, the following tweets, some long and complex while others are short and simple, are basically talking about the same thing.

1. *oh yea and Chile earthquake the earth off it's axis according to NASA and shorten the day by a second :-(*
2. *Chile Earthquake Shortened Earth Day*
3. *Chile Earthquake Shortened Day*

Intuitively, leveraging this kind of redundancy can help SRL for tweets. Take the above tweets for example. It is hard to identify *earthquake* as the A0 argument (or the agent) of *shorten* for the first tweet because of the long distance between them. However, it is relatively easy to identify the A0 relationship between *shorten* and *earthquake* for the second or third tweet. This in turn encourages SRL to guess the same relationship exists in the first one, according to the assumption that a word tends to play the same role for the same predicate in sentences with similar content. An investigation of 1,000 similar pairs shows that for 912 pairs this assumption does hold.

We are thus motivated to collectively conduct SRL on similar tweets, exploiting the redundancy to compensate for the lack of information in a tweet. Specifically, similar tweets are first grouped using a hierarchical clustering algorithm, and then are successively labeled by two SRL systems. The first system contributes some confidently labeled results, from which statistical information is collected, such as the predicate/argument/role triples that occur frequently. The second

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

[1] http://twitter.com/

[2] http://sourceforge.net/projects/opennlp/

system uses such statistical information plus other conventional information (e.g., lemma, POS, etc.) from a tweet to further analyze the predicate-argument structures. This two-stage labeling strategy, which aims to augment tweets that are hard to label by using tweets with similar content but are easy to analyze, differentiates ours from existing methods.

Notably, this two-stage labeling strategy has been successfully applied to other tasks such as event extraction [Liao and Grishman, 2010], where a sentence-level system is first used to make decisions based on local information, followed by a document level system, which deals with cases that the local system cannot handle, with the help of the confident local information collected.

To evaluate our method, 6,670 tweets are manually annotated. The modified version of our method that bypasses second round labeling is adopted as the baseline. Experimental results show that our method significantly boots the baseline by 3.1% F1.

Our contributions are summarized as follows.

1. We propose to exploit the redundancy in similar tweets to compensate for the lack of information in a tweet, a main challenge of SRL for tweets.
2. We propose a two-stage labeling strategy to exploit the redundancy in similar tweets. That is, in the first stage, the statistical information is collected from results that are labeled confidently by a system, which is used in the second stage by another system to refine the results.
3. We evaluate our method on a human annotated dataset, showing that our method remarkably improves the baseline by 3.1% F1.

The rest of our paper is organized as follows. In the next section, we introduce related work. In Section 3, we formally define the task. In Section 4, we describe our method. In Section 5, we evaluate our method. Finally, Section 6 concludes and presents future work.

2 Related Work

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

2.1 SRL for Non-tweets

Gildea and Jurafsky [2002] introduce the task of SRL. By now it has attracted increasing attention owing to its usefulness to other NLP tasks and applications, such as information extraction, question answering, and machine translation. 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 separately [Xue, 2004; Koomen *et al.*, 2005; Cohn and Blunsom, 2005; Punyakanok *et al.*, 2008; Toutanova *et al.*, 2005; 2008] .

Non-pipelined approaches exist as well. 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]; Liu et al. [2010a] simultaneously conduct SRL for multiple news sentences that are similar in content to leverage the content redundancy using the MLN framework.

All the above methods mainly focus on normal text; in contrast, our method focuses on SRL on tweets, a new genre of text. Specifically, compared with the work of Liu et al. [2010a] that uses MLN, our method adopts two-stage labeling, a very lightweight framework, to collectively label multiple similar tweets.

2.2 SRL for Tweets

Liu et al. [2010b] first study the task of SRL for tweets. They map predicate-argument structures from news sentences to news tweets (tweets that report news) to get training data, based on which a tweet specific system is trained. A linear Conditional Random Fields (CRF) model is used to integrate conventional features such as lemma and POS. There are two substantial differences between this work and ours. Firstly, Liu et al. [2010b] focus only on news tweets while ours extends their scope to general tweets. It is worth noting that, news tweets represent only a small portion of all tweets, e.g., 13.5% according to our investigation, and that these tweets are generally easier for SRL, as partially evidenced by one of our experiments in which its F1 drops from 66.0% on news tweets to 44.3% on general tweets. Secondly, the focus of their method is to automatically get training data, while ours is to utilize similar tweets to enrich the information in a tweet.

2.3 Semi-supervised Learning for SRL

Leveraging the tweets that are confidently labeled by itself in the second labeling stage, our method is essentially a kind of semi-supervised learning method. Semi-supervised learning has been widely used in scenarios where labeled data is scarce and hard to construct while unlabeled data is abundant and easy to access. Most recently, Huang et al. [2010] propose to learn latent-variable language models from a large volume of unlabeled data using Hidden Markov Models (HMMs), based on which the input text is represented. They report that this representation reduces error by 16.0% relative to the previous state-of-the-art on out-of-domain text. In contrast, our method aggregates information from similar tweets that are labeled confidently to combat the lack of information in a tweet.

3 Task Definition

We first introduce some background about tweets, then give a formal definition of the task.

3.1 The Tweets

A tweet is a short text message with no more than 140 characters in Twitter. Twitter is the most popular micro-blog service, where users use tweets to freely discuss any topic, report whatever is happening, communicate with each other, share information with each other, and follow each other. Now Twitter has become a comprehensive repository for super fresh information.

Here is an example of a tweet: "mycraftingworld: #Win Microsoft Office 2010 Home and Student *2Winners* #Contest from @office and @momtobedby8 #Giveaway http://bit.ly/bCsLOr ends 11/14", where "mycraftingworld" is the name of the user who published this tweet. Words beginning with the "#" character, like ""#Win", "#Contest" and "#Giveaway", are hash tags, usually indicating the topics of the tweet; words starting with "@", like "@office" and "@momtobedby8", represent user names, and "http://bit.ly/bCsLOr" is a shortened link.

3.2 The Task

Given a data stream of tweets, our task is to identify the predicate-argument structures for each tweet. 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). Only verbal predicates are considered, which is consistent with most existing SRL systems. Following Màrquez et al. [2005], we conduct word level labeling. As a pilot study, we restrict the scope to English tweets, though our method can be straightforwardly extended to support tweets of other languages.

Here is an example to illustrate the task. For the tweet "oh yea and Chile earthquake shorten the day by a wee second :-(". The expected output is a set of triples: {(shorten,earthquake,A0),(shorten,day,A1)}, which says that "earthquake" and "day" are the agent and patient of "shorten", respectively.

4 Our Method

Now we describe our two-stage labeling solution. An overview of our method is first given, followed by detailed discussion of its core components.

4.1 Method Overview

Algorithm 1 outlines our method, where: l_1, l_2 denote two labelers; the $cluster$ function puts a tweet into a cluster; the $label$ function generates predicate-argument structures for the input tweet; p, s and cf denote a predicate, a set of argument-role pairs related to the predicate and the predicted confidence, respectively; cs denotes the cache of confidently labeled results; N (20) and α (0.4) refer to the maximum allowable size of a cluster and the minimum allowable confidence, respectively.

From Algorithm 1, it can be seen that our method first tries to put a tweet into a cluster, and if the size of a cluster goes beyond N, all tweets in the cluster will be processed in the following way: Firstly, l_1 is applied and the labeled results with $cf > \alpha$ are cached into cs. Secondly, l_2 is applied with cs and its labeled results are outputted. Note that l_1 is used to label the unprocessed tweets when the input stream is empty.

Algorithm 1 Collective SRL for tweets with Clustering.

Require: Tweet stream i;sequential labelers l_1, l_2;output stream o.

```
Initialize clusters cl: cl = ∅.
while Pop a tweet t from i and t ≠ null do
    Put t to a cluster c: (c, cl) = cluster(cl, t).
    if |c| > N then
        Initialize cache of labeled results cs:cs = ∅.
        for ∀t' ∈ c do
            Label t' with l1:(t', {(p, s, cf)}) = label(l1, t').
            for ∀cf ∈ {(p, s, cf)} > α do
                Cache labeled results:cs = cs ∪ {(t', p, s, cf)}.
            end for
        end for
        for ∀t' ∈ c do
            Label t' with l2:(t', {(p, s, cf)}) = label(cs, l2, t').
            Output labeled results (t', {(p, s, cf)}) to o.
        end for
        Remove c from cl:cl = cl − {c}.
    end if
end while
for ∀c ∈ cl, ∀t' ∈ c do
    Label t' with l1:(t', {(p, s, cf)}) = label(l1, t').
    Output labeled results (t', {(p, s, cf)}) to o.
end for
return o.
```

4.2 Model

Both labelers are based on linear CRF models, 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; and 2) compared with its alternatives, such as those based on MLN, it is faster with comparable performance.

In line with 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 the linear CRF models, and to implement the $label$ function in Algorithm 1 as well.

4.3 Features

Before feature extraction, extracted tweet meta data is normalized so that each link and account name become LINK and ACCOUNT, respectively. Hash tags are treated as common words. Moreover, a simple dictionary-lookup based normalization procedure is conducted, using a pre-compiled list including incorrect/correct word pairs, e.g., "loooove/"love", to correct common ill-formed words.

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

For the first labeler l_1, the conventional features defined in Màrquez et al. [2005] are used, including 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. [2010b], dependencies parsing related features are used as well. The OpenNLP toolkit and the Stanford parser [4] are used to extract these features.

For the second labeler l_2, besides conventional features, another set of features derived from the cluster is used. That is, for each word, the top K (3 in our work) most frequent role-predicate pairs in the pre-labeled results are used as features. It is worth noting that, in the training stage, manually annotated tweets are clustered using Algorithm 2 so that for each tweet these features can be extracted from the tweets in the same group.

4.4 Clustering Tweets

Algorithm 2 shows the clustering process. The tweet for labeling is put into the most similar cluster if the similarity is no less than a threshold β (0.4 in our work); otherwise a new cluster is generated for this tweet. The $merge$ function, which combines the two most similar clusters into one, is called whenever the number of clusters goes beyond M (experimentally set to 1,000).

Algorithm 2 Clustering a tweet.

Require: Clusters cl;tweet for clustering t.

1: Get the reference of the most near cluster c^* for t:$c^* = argmax_{c' \in cl} sim(t, c')$.
2: Get the similarity s between t and c^*: $s = sim(t, c^*)$.
3: **if** $s < \beta$ **then**
4: Create a new cluster for t: $c^* = \{t\}$.
5: Add c^* to cl:$cl = cl \cup \{c^*\}$.
6: **if** $|cl| > M$ **then**
7: Merge clusters:$cl = merge(cl)$.
8: **end if**
9: **end if**
10: **return** c^* and cl.

To compute the similarity between a tweet and a cluster, both of them are first represented as bag-of-words vectors (Formula 1), with stop words removed and meta data normalized. Stop words are chosen mainly from a list of common words [5]; then the cosine function is applied (Formula 2).

$$\vec{c} = \sum_{\vec{t} \in c} \vec{t} \tag{1}$$

$$sim(\vec{t_1}, \vec{t_2}) = \frac{\vec{t_1} \cdot \vec{t_2}}{|\vec{t_1}||\vec{t_2}|} \tag{2}$$

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

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

4.5 Discussion

We now discuss two issues related to our method, i.e., the predication detection, and the breaking of the SRL structural constraints.

Predicate Detection. Given a tweet, finding out all possible predicates largely depends on the POS tagger. We observe that current state-of-the-art POS taggers work surprisingly well in identifying verbs. For example, the OpenNLP toolkit can achieve an F1 of 87.0% on our test dataset for verbs (in contrast with its overall accuracy of 74.0% on this dataset) [6]. To further improve the performance of POS on verbs, the Stanford tagger [Toutanova *et al.*, 2003] and a simple rule-based tagger [Brill, 1992] are combined with the OpenNLP tagger using a simple voting strategy. The final F1 for verbs is 88.6 %.

Structural Constraints. SRL introduces two structural constraints: 1) One word w can play at most one role r for a predicate p; and 2) some roles, such as A0 and A1, can have at most one word w as an argument for a predicate p. Pipelined methods adopt a post processing process to enforce such constraints, while MLN-based methods explicitly encode them using first order formulas. Systems based on sequential labeling naturally obey the first rule, but not always the second, since knowledge about the second is implicitly learnt from the training data. A question arises: How well is the second rule obeyed? We investigate the outputs of our method on the test dataset, and get only 467 (out of 3,276) cases in which the second constraint does not hold, suggesting that our linear CRF models can effectively learn this constraint from the training data.

5 Experiments

In this section, we first introduce how the experimental dataset is prepared, and then evaluate our SRL system on this dataset, and show that our method significantly outperforms the baseline that ignores redundancy in similar tweets.

5.1 Data Preparation

We use the Twitter API to crawl all tweets from April 20^{th} 2010 to April 25^{th} 2010, then drop non-English tweets and get about 11,371,389 tweets, from which 7,499 tweets are randomly sampled. The selected tweets are then labeled by 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 6,670 tweets are kept, forming the gold-standard dataset [7]. Statistical information about this dataset is presented in Figures 1-2.

The gold-standard dataset is randomly divided into three parts, for development (1,000), training (2,394) and testing (the remaining), respectively. The systematic parameters, i.e., N, M,K, α and β, are experimentally set to the optimal values, which yield the best performance on the development

[6] The POS information is manually annotated for the test dataset to support this evaluation.

[7] The Kappa agreement between the two annotators is 0.78.

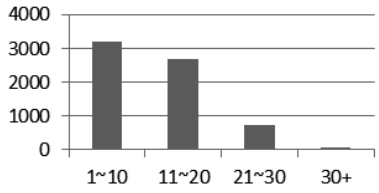

Figure 1: Tweets with a different number of words in the gold-standard dataset. Horizontal and vertical axes represent the number of words and tweets, respectively.

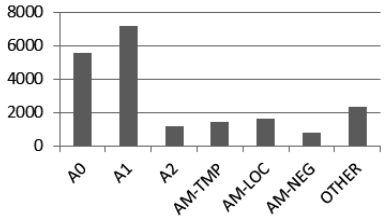

Figure 2: Arguments of different types in the gold-standard dataset. Horizontal and vertical axes represent argument type and the number of its occurrence, respectively.

dataset. Under the optimal setting, the training and the test dataset have 109 and 108 clusters, respectively.

5.2 Evaluation Metrics

Following the common practice, 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 dataset are correctly labeled, while F1 is the harmonic mean of precision and recall.

5.3 Reference System Performance

Two off-the-shelf systems are studied to understand how existing systems performs on tweets. One is the MLN based system [Meza-Ruiz and Riedel, 2009], which is trained on the CoNLL08 shared task dataset and achieves state-of-the-art performance on that task; the other is the tweet specific system from Liu et al. [Liu *et al.*, 2010b], 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) are used to extract conventional features for the reference systems. Table 1 shows the performance of these two systems and ours on the same test dataset, where SRL_{MLN}, SRL_{TN} and SRL_{CL} denote the MLN based system [Meza-Ruiz and Riedel, 2009], the system from Liu et al. [2010b] and ours, respectively. Note that all these systems conduct word level SRL. From Table 1, it can been seen that ours performs remarkably better than SRL_{MLN} and SRL_{TN} . This is understandable since SRL_{MLN} is trained on formal text while SRL_{TN} is trained on mechanically annotated news tweets.

The test dataset from Liu et al. [2010b] is used as well to evaluate our method. The F1 is 65.1%, almost as good as that of SRL_{TN} (66.0%). This can be explained by the fact that our method, though not trained on news tweets, uses human labeled tweets and two-stage labeling. In future, we plan to apply our method to Liu et al. [2010b], to see if it works on automatically labeled training data.

Table 1: Performances of reference systems.

System	Pre.(%)	Rec.(%)	F1(%)
SRL_{CL}	61.9	56.7	59.2
SRL_{TN}	54.1	37.5	44.3
SRL_{MLN}	40.0	49.9	44.4

Table 2: Comparison to the baseline.

System	Pre.(%)	Rec.(%)	F1(%)
SRL_{CL}	61.9	56.7	59.2
SRL_{BA}	62.7	50.8	56.1

5.4 Baseline and Results

A modified version of our method that uses only l_1, hereafter denoted by SRL_{BA}, is adopted as the baseline. Table 2 shows the experimental results of the baseline and ours, respectively. From Table 2, it can be seen that the second round labeling significantly boosts the F1 from 56.1% to 59.2% (with $p < 0.03$), suggesting the contribution of redundancy in similar tweets. Table 3 presents detailed results of our method for different roles.

5.5 Error Analysis

A manual check reveals that more than 60.2% of errors made by our system are owing to 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 no punctuation follows "@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". We are developing a tweet specific POS system and advanced tweet normalization technologies to correct these errors.

Another great part of errors, which accounts for 39.8% of all errors, is linked to the fact that training data cannot fully encode the knowledge about SRL for tweets, which are diversified in linguistic realizations and noise prone. For example,

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

Role	Pre.(%)	Rec.(%)	F1(%)
A0	73.0	75.7	74.3
A1	58.0	52.3	55.0
A2	41.6	31.0	35.5
AM-TMP	49.6	39.3	43.9
AM-LOC	56.2	53.5	54.8
AM-NEG	86.7	73.2	79.4
OTHER	44.5	43.1	43.8

our system cannot correctly label this tweet "Bacteria in the 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 or using bootstrapping seems a promising way to fix this kind of error.

6 Conclusions and Future work

The task of SRL for tweets is challenging, because a tweet is often too short and informal to provide sufficient information. We propose a two-stage labeling method that leverages similar tweets to combat this challenge. Experimental results show that our method can achieve an absolute F1 gain of 3.1%. We are developing POS systems and advanced normalization technologies for tweets to further improve our method.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. We also thank Matt Callcut for his careful proofreading of an early draft of this paper.

References

[Brill, 1992] Eric Brill. A simple rule-based part of speech tagger. In *HLT*, pages 112–116, 1992.

[Carreras and Màrquez, 2005] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *CoNLL*, pages 152–164, 2005.

[Cohn and Blunsom, 2005] Trevor Cohn and Philip Blunsom. Semantic role labelling with tree conditional random fields. In *CONLL*, pages 169–172, 2005.

[Dong *et al.*, 2010] Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. Time is of the essence: improving recency ranking using twitter data. In *WWW*, pages 331–340, 2010.

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

[Huang and Yates, 2010] Fei Huang and Alexander Yates. Open-domain semantic role labeling by modeling word spans. In *ACL*, pages 968–978, 2010.

[Kingsbury and Palmer, 2003] Paul Kingsbury and Martha Palmer. Propbank: The next level of treebank. In *Proceedings of Treebanks and Lexical Theories*, 2003.

[Koomen *et al.*, 2005] Peter Koomen, Vasin Punyakanok, Dan Roth, and Wen-tau Yih. Generalized inference with multiple semantic role labeling systems. In *CONLL*, pages 181–184, 2005.

[Liao and Grishman, 2010] Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In *ACL*, pages 789–797, 2010.

[Liu *et al.*, 2010a] Xiaohua Liu, Kuan Li, Bo Han, Ming Zhou, Long Jiang, Daniel Tse, and Zhongyang Xiong. Collective semantic role labeling on open news corpus by leveraging redundancy. In *Coling*, pages 725–729, 2010.

[Liu *et al.*, 2010b] Xiaohua Liu, Kuan Li, Bo Han, Ming Zhou, Long Jiang, Zhongyang Xiong, and Changning Huang. Semantic role labeling for news tweets. In *Coling*, pages 698–706, 2010.

[Màrquez *et al.*, 2005] Lluís Màrquez, Pere Comas, Jesús Giménez, and Neus Català. Semantic role labeling as sequential tagging. In *CONLL*, pages 193–196, 2005.

[Merlo and Musillo, 2008] Paola Merlo and Gabriele Musillo. Semantic parsing for high-precision semantic role labelling. In *CoNLL*, pages 1–8, 2008.

[Meza-Ruiz and Riedel, 2009] Ivan Meza-Ruiz and Sebastian Riedel. Jointly identifying predicates, arguments and senses using markov logic. In *NAACL*, pages 155–163, 2009.

[Musillo and Merlo, 2006] Gabriele Musillo and Paola Merlo. Accurate parsing of the proposition bank. In *NAACL*, pages 101–104, 2006.

[Punyakanok *et al.*, 2008] Vasin Punyakanok, Dan Roth, and Wen-tau Yih. The importance of syntactic parsing and inference in semantic role labeling. *Comput. Linguist.*, 34:257–287, 2008.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach. Learn.*, 62:107–136, 2006.

[Sankaranarayanan *et al.*, 2009] Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *GIS*, pages 42–51, 2009.

[Surdeanu *et al.*, 2008] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL*, pages 159–177, 2008.

[Toutanova *et al.*, 2003] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*, pages 173–180, 2003.

[Toutanova *et al.*, 2005] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. Joint learning improves semantic role labeling. In *ACL*, pages 589–596, 2005.

[Toutanova *et al.*, 2008] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. A global joint model for semantic role labeling. *Comput. Linguist.*, 34:161–191, 2008.

[Vickrey and Koller, 2008] David Vickrey and Daphne Koller. Applying sentence simplification to the conll-2008 shared task. In *CoNLL*, pages 268–272, 2008.

[Xue, 2004] Nianwen Xue. Calibrating features for semantic role labeling. In *EMNLP*, pages 88–94, 2004.