Multilingual Syntactic-Semantic Dependency Parsing with Three-Stage Approximate Max-Margin Linear Models

Yotaro Watanabe, Masayuki Asahara and Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, Japan, 630-0192
{yotaro-w, masayu-a, matsu}@is.naist.jp

Abstract

This paper describes a system for syntactic-semantic dependency parsing for multiple languages. The system consists of three parts: a state-of-the-art higher-order projective dependency parser for syntactic dependency parsing, a predicate classifier, and an argument classifier for semantic dependency parsing. For semantic dependency parsing, we explore use of global features. All components are trained with an approximate max-margin learning algorithm.

In the closed challenge of the CoNLL-2009 Shared Task (Hajič et al., 2009), our system achieved the 3rd best performances for English and Czech, and the 4th best performance for Japanese.

1 Introduction

In recent years, joint inference of syntactic and semantic dependencies has attracted attention in NLP communities. Ideally, we would like to choose the most plausible syntactic-semantic structure among all possible structures in that syntactic dependencies and semantic dependencies are correlated. However, solving this problem is too difficult because the search space of the problem is extremely large. Therefore we focus on improving performance for each subproblem: dependency parsing and semantic role labeling.

In the past few years, research investigating higher-order dependency parsing algorithms has found its superiority to first-order parsing algorithms. To reap the benefits of these advances, we use a higher-order projective dependency parsing algorithm (Carreras, 2007) which is an extension of the span-based parsing algorithm (Eisner, 1996), for syntactic dependency parsing.

In terms of semantic role labeling, we would like to capture global information about predicate-argument structures in order to accurately predict the correct predicate-argument structure. Previous research dealt with such information using *re-ranking* (Toutanova et al., 2005; Johansson and Nugues, 2008). We explore a different approach to deal with such information using *global features*. Use of global features for structured prediction problem has been explored by several NLP applications such as sequential labeling (Finkel et al., 2005; Krishnan and Manning, 2006; Kazama and Torisawa, 2007) and dependency parsing (Nakagawa, 2007) with a great deal of success. We attempt to use global features for argument classification in which the most plausible semantic role assignment is selected using both local and global information. We present an approximate max-margin learning algorithm for argument classifiers with global features.

2 Dependency Parsing

As in previous work, we use a linear model for dependency parsing. The score function used in our dependency parser is defined as follows.

$$s(\mathbf{y}) = \sum_{(h,m) \in \mathbf{y}} F(h, m, \mathbf{x}) \qquad (1)$$

where h and m denote the head and the dependent of the dependency edge in $\mathbf{y}$, and $F(h, m, \mathbf{x})$ is a *Factor* that specifies dependency edge scores.

We used a second-order factorization as in (Carreras, 2007). The second-order factor F is defined as follows.

$$F(h, m, \mathbf{x}) = \mathbf{w} \cdot \Phi(h, m, \mathbf{x}) + \mathbf{w} \cdot \Phi(h, m, c_h, \mathbf{x})$$
$$+ \mathbf{w} \cdot \Phi(h, m, c_{mi}, \mathbf{x}) + \mathbf{w} \cdot \Phi(h, m, c_{mo}, \mathbf{x}) \quad (2)$$

where $\mathbf{w}$ is a parameter vector, Φ is a feature vector, c_h is the child of h in the span $[h...m]$ that is closest to m, c_{mi} is the child of m in the span $[h...m]$ that is farthest from m and c_{mo} is the child of m outside the span $[h...m]$ that is farthest from m. For more details of the second-order parsing algorithm, see (Carreras, 2007).

For parser training, we use the Passive Aggressive Algorithm (Crammer et al., 2006), which is an approximate max-margin variant of the perceptron algorithm. Also, we apply an efficient parameter averaging technique (Daumé III, 2006). The resulting learning algorithm is shown in Algorithm 1.

Algorithm 1 A Passive Aggressive Algorithm with parameter averaging

---

input Training set $\mathcal{T} = \{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^T$, Number of iterations N and Parameter C
$\mathbf{w} \leftarrow \mathbf{0}, \mathbf{v} \leftarrow \mathbf{0}, c \leftarrow 1$
for $i \leftarrow 0$ to N **do**
  for $(\mathbf{x}_t, \mathbf{y}_t) \in \mathcal{T}$ **do**
    $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathbf{y}) + \rho(\mathbf{y}_t, \hat{\mathbf{y}})$
    $\tau_t = \min\left(C, \frac{\mathbf{w} \cdot \Phi(\mathbf{x}_t, \hat{\mathbf{y}}) - \mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathbf{y}_t) + \rho(\mathbf{y}_t, \hat{\mathbf{y}})}{||\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}})||^2}\right)$
    $\mathbf{w} \leftarrow \mathbf{w} + \tau_t(\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}))$
    $\mathbf{v} \leftarrow \mathbf{v} + c\tau_t(\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}))$
    $c \leftarrow c + 1$
  end for
end for
return $\mathbf{w} - \mathbf{v}/c$

We set $\rho(\mathbf{y}_t, \hat{\mathbf{y}})$ as the number of incorrect head predictions in the $\hat{\mathbf{y}}$, and C as 1.0.

Among the 7 languages of the task, 4 languages (Czech, English, German and Japanese) contain non-projective edges (13.94 %, 3.74 %, 25.79 % and 0.91 % respectively), therefore we need to deal with non-projectivity. In order to avoid losing the benefits of higher-order parsing, we considered applying pseudo-projective transformation (Nivre and Nilsson, 2005). However, growth of the number of dependency labels by pseudo-projective transformation increases the dependency parser training time, so we did not adopt transformations. Therefore, the

parser ignores the presence of non-projective edges in the training and the testing phases.

The features used for our dependency parser are based on those listed in (Johansson, 2008). In addition, distance features are used. We use shorthand notations in order to simplify the feature representations: 'h', 'd', 'c', 'l', 'p', '−1' and '+1' correspond to head, dependent, head's or dependent's child, lemma , POS, left position and right position respectively.

First-order Features

Token features: hl, hp, hl+hp, dl, dp and dl+dp.

Head-Dependent features: hp+dp, hl+dl, hl+dl, hl+hp+dl, hl+hp+dp, hl+dl+dp, hp+dl+dp and hl+hp+dl+dp.

Context features: hp+hp$_{+1}$+dp$_{-1}$+dp, hp$_{-1}$+hp+dp$_{-1}$+dp, hp+hp$_{+1}$+dp+dp$_{+1}$ and hp$_{-1}$+hp+dp+dp$_{+1}$.

Distance features: The number of tokens between the head and the dependent.

Second-order Features

Head-Dependent-Head's or Dependent's Child: hl+cl, hl+cl+cp, hp+cl, hp+cp, hp+dp+cp, dp+cp, dp+cl+cp, dl+cp, dl+cp+cl

3 Semantic Role Labeling

Our SRL module consists of two parts: a predicate classifier and an argument classifier. First, our system determines the word sense for each predicate with the predicate classifier, and then it detects the highest scored argument assignment using the argument classifier with global features.

3.1 Predicate Classification

The first phase of SRL in our system is to detect the word sense for each predicate. WSD can be formalized as a multi-class classification problem given lemmas. We created a linear model for each lemma and used the Passive Aggressive Algorithm with parameter averaging to train the models.

3.1.1 Features for Predicate Classification

Word features: Predicted lemma and the predicted POS of the predicate, predicate's head, and its conjunctions.

Dependency label: The dependency label between the predicate and the predicate's head.

Dependency label sequence: The concatenation of the dependency labels of the predicate dependents.

Since effective features for predicate classification are different for each language, we performed greedy forward feature selection.

3.2 Argument Classification

In order to capture global clues of predicate-argument structures, we consider introducing global features for linear models. Let $\mathcal{A}(p)$ be a joint assignment of role labels for argument candidates given the predicate p. Then we define a score function $s(\mathcal{A}(p))$ for argument label assignments $\mathcal{A}(p)$.

$$s(\mathcal{A}(p)) = \sum_k F_k(\mathbf{x}, \mathcal{A}(p)) \tag{3}$$

We introduce two factors: *Local Factor* F_L and *Global Factor* F_G defined as follows.

$$F_L(\mathbf{x}, a(p)) = \mathbf{w} \cdot \Phi_L(\mathbf{x}, a(p)) \tag{4}$$

$$F_G(\mathbf{x}, \mathcal{A}(p)) = \mathbf{w} \cdot \Phi_G(\mathbf{x}, \mathcal{A}(p)) \tag{5}$$

where Φ_L, Φ_G denote feature vectors for the local factor and the global factor respectively. F_L scores a particular role assignment for each argument candidate individually, and F_G treats global features that capture what structure the assignment $\mathcal{A}$ has. Resulting scoring function for the assignment $\mathcal{A}(p)$ is as follows.

$$s(\mathcal{A}(p)) = \sum_{a(p) \in \mathcal{A}(p)} \mathbf{w} \cdot \Phi_L(\mathbf{x}, a(p)) + \mathbf{w} \cdot \Phi_G(\mathbf{x}, \mathcal{A}(p)) \tag{6}$$

Use of global features is problematic, because it becomes difficult to find the highest assignment efficiently. In order to deal with the problem, we use a simple approach, n-best relaxation as in (Kazama and Torisawa, 2007). At first we generate n-best assignments using only the local factor, and then add the global factor score for each n-best assignment, finally select the best scoring assignment from them. In order to generate n-best assignments, we used a beam-search algorithm.

3.2.1 Learning the Model

As in dependency parser and predicate classifier, we train the model using the PA algorithm with parameter averaging. The learning algorithm is shown in Algorithm 2. In this algorithm, the weights correspond to local factor features Φ_L and global factor features Φ_G are updated simultaneously.

Algorithm 2 Learning with Global Features for Argument Classification

---

input Training set $\mathcal{T} = \{\mathbf{x_t}, \mathcal{A}_t\}_{t=1}^{T}$, Number of iterations N and Parameter C
$\mathbf{w} \leftarrow \mathbf{0}, \mathbf{v} \leftarrow \mathbf{0}, c \leftarrow 1$
for $i \leftarrow 0$ to N **do**
    for $(\mathbf{x}_t, \mathcal{A}_t) \in \mathcal{T}$ **do**
        let $\Phi(\mathbf{x}_t, \mathcal{A}) = \sum_{a \in \mathcal{A}} \Phi_L(\mathbf{x}_t, a) + \Phi_G(\mathbf{x}_t, \mathcal{A})$
        generate n-best assignments $\{\mathcal{A}^n\}$ using F_L
        $\hat{\mathcal{A}} = \arg\max_{\mathcal{A} \in \{\mathcal{A}^n\}} \mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathcal{A}) + \rho(\mathcal{A}_t, \mathcal{A})$
        $\tau_t = \min\left(C, \frac{\mathbf{w} \cdot \Phi(\mathbf{x}_t, \hat{\mathcal{A}}) - \mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathcal{A}_t) + \rho(\mathcal{A}_t, \hat{\mathcal{A}})}{||\Phi(\mathbf{x}_t, \mathcal{A}_t) - \Phi(\mathbf{x}_t, \hat{\mathcal{A}})||^2}\right)$
        $\mathbf{w} \leftarrow \mathbf{w} + \tau_t(\Phi(\mathbf{x}_t, \mathcal{A}_t) - \Phi(\mathbf{x}_t, \hat{\mathcal{A}}))$
        $\mathbf{v} \leftarrow \mathbf{v} + c\tau_t(\Phi(\mathbf{x}_t, \mathcal{A}_t) - \Phi(\mathbf{x}_t, \hat{\mathcal{A}}))$
        $c \leftarrow c + 1$
    end for
end for
return $\mathbf{w} - \mathbf{v}/c$

We set the margin value $\rho(\mathcal{A}, \hat{\mathcal{A}})$ as the number of incorrect assignments plus $\delta(\mathcal{A}, \hat{\mathcal{A}})$, and C as 1.0. The delta function returns 1 if at least one assignment is different from the correct assignment and 0 otherwise.

The model is similar to re-ranking (Toutanova et al., 2005; Johansson and Nugues, 2008). However in contrast to re-ranking, we only have to prepare one model. The re-ranking approach requires other training datasets that are different from the data used in local model training.

3.2.2 Features for Argument Classification

The local features used in our system are the same as our previous work (Watanabe et al., 2008) except for language dependent features. The global features that used in our system are based on (Johansson and Nugues, 2008) that used for re-ranking.

Local Features

Word features: Predicted lemma and predicted POS of the predicate, predicate's head, argument candidate, argument candidate's head, leftmost/rightmost dependent and leftmost/rightmost sibling.

Dependency label: The dependency label of predicate, argument candidate and argument candidate's dependent.

Family: The position of the argument candidate with respect to the predicate position in the dependency tree (e.g. child, sibling).

	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
Macro F1 Score	78.43	75.91	73.43	81.43	86.40	69.84	84.86	77.12
	(78.00*)	(74.83*)	(73.43*)	(81.38*)	(86.40*)	(68.39*)	(84.84*)	(76.74*)
Semantic Labeled F1	75.65	72.35	74.17	84.69	84.26	63.66	77.93	72.50
	(75.17*)	(71.05*)	(74.17*)	(84.66*)	(84.26*)	(61.94*)	(77.91*)	(72.25*)
Labeled Syntactic Accuracy	81.16	79.48	72.66	78.17	88.54	75.85	91.69	81.74
	(80.77*)	(78.62*)	(72.66*)	(78.10*)	(88.54*)	(74.60*)	(91.66*)	(81.23*)
Macro F1 Score	84.30	84.79	81.63	83.08	87.93	83.25	85.54	83.94
Semantic Labeled F1	81.58	80.99	79.99	86.67	85.09	79.46	79.03	79.85
Labeled Syntactic Accuracy	87.02	88.59	83.27	79.48	90.77	87.03	91.96	88.04

Table 1: Scores of our system.

Position: The position of the head of the dependency relation with respect to the predicate position in the sentence.

Pattern: The left-to-right chain of the predicted POS/dependency labels of the predicate's children.

Path features: Predicted lemma, predicted POS and dependency label paths between the predicate and the argument candidate.

Distance: The number of dependency edges between the predicate and the argument candidate.

Global Features

Predicate-argument label sequence: The sequence of the predicate sense and argument labels in the predicate-argument strucutre.

Presence of labels defined in frame files: Whether the semantic roles defined in the frame present in the predicate-argument structure (e.g. MISSING:A1 or CONTAINS:A1.)

3.2.3 Argument Pruning

We observe that most arguments tend to be not far from its predicate, so we can prune argument candidates to reduce search space. Since the characteristics of the languages are slightly different, we apply two types of pruning algorithms.

Pruning Algorithm 1: Let S be an argument candidate set. Initially set $S \leftarrow \phi$ and start at predicate node. Add dependents of the node to S, and move current node to its parent. Repeat until current node reaches to ROOT.

Pruning Algorithm 2: Same as the Algorithm 1 except that added nodes are its grandchildren as well as its dependents.

The pruning results are shown in Table 2. Since we could not prune arguments in Japanese accurately using the two algorithms, we pruned argument candidates simply by POS.

	algorithm	coverage (%)	reduction (%)
Catalan	1	100	69.1
Chinese	1	98.9	69.1
Czech	2	98.5	49.1
English	1	97.3	63.1
German	1	98.3	64.3
Japanese	POS	99.9	41.0
Spanish	1	100	69.7

Table 2: Pruning results.

4 Results

The submitted results on the test data are shown in the upper part of Table 1. Due to a bug, we mistakenly used the gold lemmas in the dependency parser. Corrected results are shown in the part marked with *. The lower part shows the post evaluation results with the gold lemmas and POSs.

For some of the 7 languages, since the global model described in Section 3.2 degraded performance compare to a model trained with only F_L, we did **NOT** use the model for all languages. We used the global model for only three languages: Chinese, English and Japanese. The remaining languages (Catalan, Czech, German and Spanish) used a model trained with only F_L.

4.1 Dependency Parsing Results

The parser achieved relatively high accuracies for Czech, English and Japanese, and for each language, the difference between the performance with correct POS and predicted POS is not so large. However, in Catalan, Chinese German and Spanish, the parsing accuracies was seriously degraded by replacing correct POSs with predicted POSs (6.3 - 11.2 %). This is likely because these languages have relatively low predicted POS accuracies (92.3 - 95.5 %) ; Chinese

	F_L	F_L+F_G	(ΔP, ΔR)
Catalan	**85.80**	85.68	(+0.01, -0.26)
Chinese	86.58	**87.39**	(+0.24, +1.36)
Czech	**89.63**	89.05	(-0.87, -0.28)
English	85.66	**85.74**	(-0.87, +0.98)
German	**80.82**	77.30	(-7.27, +0.40)
Japanese	79.87	**81.01**	(+0.17, +1.88)
Spanish	**84.38**	83.89	(-0.42, -0.57)

Table 3: Effect of global features (semantic labeled F1). ΔP and ΔR denote the differentials of labeled precision and labeled recall between F_L and F_L+F_G respectively.

	iter	hrs./iter	sent./min.	mem.
Catalan	9	14.6	9.0	9.6 GB
Chinese	2	56.5	3.7	16.2 GB
Czech	8	14.6	20.5	12.6 GB
English	7	22.0	13.4	15.1 GB
German	4	12.3	59.1	13.1 GB
Japanese	7	11.2	21.8	13.0 GB
Spanish	7	19.5	7.3	17.9 GB

Table 4: Training, evaluation time and memory requirements of the second-order dependency parsers. The 'iter' column denote the number of iterations of the model used for the evaluations. Catalan, Czech and English are trained on Xeon 3.0GHz, Chinese and Japanese are trained on Xeon 2.66GHz, German and Spanish are trained on Opteron 2.3GHz machines.

	train (hrs.)	sent./min.	mem.
Chinese	6.5	453.7	2.0 GB
English	13.5	449.8	3.2 GB
Japanese	3.5	137.6	1.1 GB

Table 5: Training, evaluation time and memory requirements of the global argument classifiers. The classifiers are all trained on Opteron 2.3GHz machines.

has especially low accuracy (92.3%). The POS accuracy may affect the parsing performances.

4.2 SRL Results

In order to highlight the effect of the global features, we compared two models. The first model is trained with only the local factor F_L. The second model is trained with both the local factor F_L and the global factor F_G. The results are shown in Table 3. In the experiments, we used the development data with gold parse trees. For Chinese and Japanese, significant improvements are obtained using the global features (over +1.0% in labeled recall and the slightly better labeled precision). However, for Catalan, Czech, German and Spanish, the global features degraded the performance in labeled F1. Especially, in German, the precision is substantially degraded (-7.27% in labeled F1). These results indicate that it is necessary to introduce language dependent features.

4.3 Training, Evaluation Time and Memory Requirements

Table 4 and 5 shows the training/evaluation times and the memory consumption of the second-order dependency parsers and the global argument classifiers respectively. The training times of the predicate classifier were less than one day, and the testing times were mere seconds.

As reported in (Carreras, 2007; Johansson and Nugues, 2008), training and inference of the second-order parser are very expensive. For Chinese, we could only complete 2 iterations.

In terms of the argument classifier, since N-best generation time account for a substantial proportion of the training time (in this work $N = 100$), chang-

ing N affects the training and evaluation times significantly.

All modules of our system are implemented in Java. The required memory spaces shown in Table 4 and 5 are calculated by subtracting free memory size from the total memory size of the Java VM. Note that we observed that the value fluctuated drastically while measuring memory usage, so the value may not indicate precise memory requirements of our system.

5 Conclusion

In this paper, we have described our system for syntactic and semantic dependency analysis in multilingual. Although our system is not a *joint* approach but a *pipeline* approach, the system is comparable to the top system for some of the 7 languages.

A further research direction we are investigating is the application of various types of global features. We believe that there is still room for improvements since we used only two types of global features for the argument classifier.

Another research direction is investigating joint approaches. To the best of our knowledge, three

types of joint approaches have been proposed: N-best based approach (Johansson and Nugues, 2008), synchronous joint approach (Henderson et al., 2008), and a joint approach where parsing and SRL are performed simultaneously (Lluís and Màrquez, 2008). We attempted to perform N-best based joint approach, however, the expensive computational cost of the 2nd-order projective parser discouraged it. We would like to investigate syntactic-semantic joint approaches with reasonable time complexities.

Acknowledgments

References

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The SALSA corpus: a German corpus resource for lexical semantics. In *Proc. of LREC-2006*, Genoa, Italy.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proc. of EMNLP-CoNLL 2007*.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *JMLR*, 7:551–585.

Hal Daumé III. 2006. *Practical Structured Learning Techniques for Natural Language Processing*. Ph.D. thesis, University of Southern California, Los Angeles, CA, August.

Jason Eisner. 1996. Three new probabilistic models for dependency parsing. In *Proc. of ICCL 1996*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL 2005*.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, and Zdeněk Žabokrtský. 2006. Prague Dependency Treebank 2.0.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proc. of CoNLL-2009*, Boulder, Colorado, USA.

James Henderson, Paola Merlo, Gabriele Musillo, and Ivan Titov. 2008. A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proc. of CoNLL 2008*.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *Proc. of CoNLL 2008*.

Richard Johansson. 2008. *Dependency-based Semantic Analysis of Natural-language Text*. Ph.D. thesis, Lund University.

Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proc. of LREC-2002*, pages 2008–2013, Las Palmas, Canary Islands.

Jun'Ichi Kazama and Kentaro Torisawa. 2007. A new perceptron algorithm for sequence labeling with non-local features. In *Proc. of EMNLP-CoNLL 2007*.

Vijay Krishnan and Christopher D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proc. of ACL-COLING 2006*.

Xavier Lluís and Lluís Màrquez. 2008. A joint model for parsing syntactic and semantic dependencies. In *Proc. of CoNLL 2008*.

Tetsuji Nakagawa. 2007. Multilingual dependency parsing using global features. In *Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proc. of ACL 2005*.

Martha Palmer and Nianwen Xue. 2009. Adding semantic roles to the Chinese Treebank. *Natural Language Engineering*, 15(1):143–172.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of CoNLL-2008*.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proc. of LREC-2008*, Marrakesh, Morroco.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *Proc. of ACL 2005*.

Yotaro Watanabe, Masakazu Iwatate, Masayuki Asahara, and Yuji Matsumoto. 2008. A pipeline approach for syntactic and semantic dependency parsing. In *Proc. of CoNLL 2008*.