Parsing Syntactic and Semantic Dependencies for Multiple Languages with A Pipeline Approach

Han Ren, Donghong Ji
School of Computer Science
Wuhan University
Wuhan 430079, China
cslotus@mail.whu.edu.cn
donghong_ji@yahoo.com

Jing Wan, Mingyao Zhang
Center for Study of Language & Information
Wuhan University
Wuhan 430079, China
{jennifer.wanj, my.zhang}@gmail.com

Abstract

This paper describes a pipelined approach for CoNLL-09 shared task on joint learning of syntactic and semantic dependencies. In the system, we handle syntactic dependency parsing with a transition-based approach and utilize MaltParser as the base model. For SRL, we utilize a Maximum Entropy model to identify predicate senses and classify arguments. Experimental results show that the average performance of our system for all languages achieves 67.81% of macro F1 Score, 78.01% of syntactic accuracy, 56.69% of semantic labeled F1, 71.66% of macro precision and 64.66% of micro recall.

1 Introduction

Given a sentence with corresponding part-of-speech for each word, the task of syntactic and semantic dependency parsing contains two folds: (1) identifying the syntactic head of each word and assigning the dependency relationship between the word and its head; (2) identifying predicates with proper senses and labeling semantic dependencies for them.

For data-driven syntactic dependency parsing, many approaches are based on supervised learning using treebank or annotated datasets. Currently, graph-based and transition-based algorithms are two dominating approaches that are employed by many researchers, especially in previous CoNLL shared tasks. Graph-based algorithms (Eisner, 1996; McDonald et al., 2005) assume a series of dependency tree candidates for a sentence and the goal is to find the dependency tree with highest score. Transition-based algorithms (Yamada and Matsumoto, 2003; Nivre et al., 2004) utilize transition histories learned from dependencies within sentences to predict next state transition and build the optimal transition sequence. Although different strategies were considered, two approaches yielded comparable results at previous tasks.

Semantic role labeling contains two problems: identification and labeling. Identification is a binary classification problem, and the goal is to identify annotated units in a sentence; while labeling is a multi-class classification problem, which is to assign arguments with appropriate semantic roles. Hacioglu (2004) utilized predicate-argument structure and map dependency relations to semantic roles. Liu et al. (2005) combined two problems into a classification one, avoiding some annotated units being excluded due to some incorrect identification results. In addition, various features are also selected to improve accuracy of SRL.

In this paper, we propose a pipelined approach for CoNLL-09 shared task on joint learning of syntactic and semantic dependencies, and describe our system that can handle multiple languages. In the system, we handle syntactic dependency parsing with a transition-based approach. For SRL, we utilize Maximum Entropy model to identify predicate senses and classify arguments.

The remain of the paper is organized as follows. In Section 2, we discuss the processing mechanism containing syntactic and semantic dependency parsing of our system in detail. In Section 3, we give the evaluation results and analysis. Finally, the conclusion and future work are given in Section 4.

2 System Description

The system, which is a two-stage pipeline, processes syntactic and semantic dependencies respectively. To reduce the difficulties in SRL, predicates of each sentence in all training and evaluation data are labeled, thus predicate identification can be ignored.
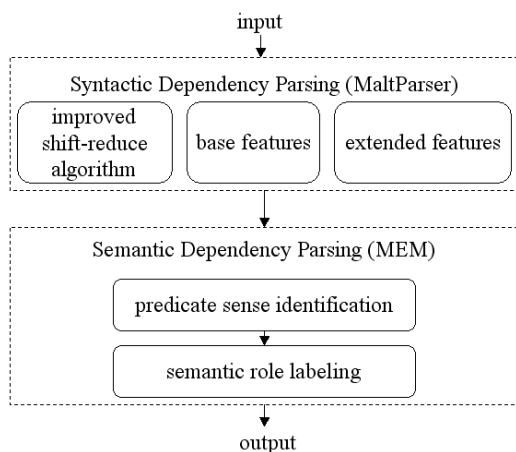

Figure 1. System Architectures

For syntactic dependencies, we employ a state-of-the-art dependency parser and basic plus extended features for parsing. For semantic dependencies, a Maximum Entropy Model is used both in predicate sense identification and semantic role labeling. Following subsections will show components of our system in detail.

2.1 Syntactic Dependency Parsing

In the system, MaltParser[1] is employed for syntactic dependency parsing. MaltParser is a data-driven deterministic dependency parser, based on a Support Vector Machine classifier. An extensive research (Nivre, 2007) parsing with 9 different languages shows that the parser is language-independent and yields good results.

MaltParser supports two kinds of parsing algorithms: Nivre's algorithms and Covington's incremental algorithms. Nivre's algorithms, which are deterministic algorithms consisting of a series of shift-reduce procedures, defines four operations:

• **Right**. For a given triple $<t|S, n|I, A>$, S represents STACK and I represents INPUT. If dependency relation $t \rightarrow n$ exists, it will be

pendency relation $t \rightarrow n$ exists, it will be appended into A and t will be removed from S.

• **Left**. For a given triple $<t|S, n|I, A>$, if dependency relation $n \rightarrow t$ exists, it will be appended into A and n will be pushed into S.

• **Reduce**. If dependency relation $n \rightarrow t$ does not exist, and the parent node of t exists left to it, t will be removed from S.

• **Shift**. If none of the above satisfies, n will be pushed into S.

The deterministic algorithm simplifies determination for Reduce operation. As a matter of fact, some languages, such as Chinese, have more flexible word order, and some words have a long distance with their children. In this case, t should not be removed from S, but be handled with Shift operation. Otherwise, dependency relations between t and its children will never be identified, thus sequential errors of dependency relations may occur after the Reduce operation.

For syntactic dependencies with long distance, an improved Reduce strategy is: if the dependency relation between n and t does not exist, and the parent node of t exists left to it and the dependency relation between the parent node and n, t will be removed from S. The Reduce operation is projective, since it doesn't influence the following parsing procedures. The Improved algorithm is described as follows:

(1) one of the four operations is performed according to the dependency relation between t and n until EOS; if only one token remains in S, go to (3).

(2) continue to select operations for remaining tokens in S; when Shift procedure is performed, push t to S; if only one token remains in S and I contains more tokens than only EOS, goto (1).

(3) label all odd tokens in S as ROOT, pointing to EOS.

We also utilize history-based feature models implemented in the parser to predict the next action in the deterministic derivation of a dependency structure. The parser provides some default features that is general for most languages: (1) part-of-speech features of TOP and NEXT and following 3 tokens; (2) dependency features of TOP containing leftmost and rightmost dependents, and of NEXT containing leftmost dependents; (3) Lexical

features of TOP, head of TOP, NEXT and following one token. We also extend features for multiple languages: (1) count of part-of-speech features of following tokens extend to 5; (2) part-of-speech and dependent features of head of TOP.

2.2 Semantic Dependency Parsing

Each defacto predicate in training and evaluation data of CoNLL09 is labeled with a sign 'Y', which simplifies the work of semantic dependency parsing. In our system, semantic dependency parsing is a pipeline that contains two parts: predicate sense identification and semantic role labeling. For predicate sense identification, each predicate is assigned a certain sense number. For semantic role labeling, local and global features are selected. Features of each part are trained by a classification algorithm. Both parts employ a Maximum Entropy Tool MaxEnt in a free package OpenNLP[2] as a classifier.

2.2.1 Predicate Sense Identification

The goal of predicate sense identification is to decide the correct frame for a predicate. According to PropBank (Palmer, et al., 2005), predicates contain one or more rolesets corresponding to different senses. In our system, a classifier is employed to identify each predicate's sense.

Suppose $C = \{01, 02, \ldots, N_L\}$ is the sense set (N_L is the count of categories corresponding to the language L, eg., in Chinese training set $N_L = 10$ since predicates have at most 10 senses in the set), and t_i is the ith sense of word w in sentence s. The model is implemented to assign each predicate to the most probatilistic sense.

$$t = argmax_{i \in C} P(w \mid s, \ t_i) \qquad (1)$$

Features for predicate sense identification are listed as follows:

• WORD, LEMMA, DEPREL: The lexical form and lemma of the predicate; the dependency relation between the predicate and its head; for Chinese and Japanese, WORD is ignored.

• HEAD_WORD, HEAD_POS: The lexical form and part-of-speech of the head of the predicate.

• CHILD_WORD_SET, CHILD_POS_SET, CHILD_DEP_SET: The lexical form, part-of-speech and dependency relation of dependents of the predicate.

• LSIB_WORD, LSIB_POS, LSIB_DEPREL, RSIB_WORD, RSIB_POS, RSIB_DEPREL: The lexical form, part-of-speech and dependency relation of the left and right sibling token of the predicate. Features of sibling tokens are adopted, because senses of some predicates can be inferred from its left or right sibling.

For English data set, we handle verbal and nominal predicates respectively; for other languages, we handle all predicates with one classifier. If a predicate in the evaluation data does not exist in the training data, it is assigned the most frequent sense label in the training data.

2.2.2 Semantic Role Labeling

Semantic role labeling task contains two parts: argument identification and argument classification. In our system the two parts are combined as one classification task. Our reason is that those argument candidates that potentially become semantic roles of corresponding predicates should not be pruned by incorrect argument identification. In our system, a predicate-argument pair consists of any token (except predicates) and any predicate in a sentence. However, we find that argument classification is a time-consuming procedure in the experiment because the classifier spends much time on a great many of invalid predicate-argument pairs. To reduce useless computing, we add a simple pruning method based on heuristic rules to remove invalid pairs, such as punctuations and some functional words.

Features used in our system are based on (Hacioglu, 2004) and (Pradhan et al, 2005), and described as follows:

• WORD, LEMMA, DEPREL: The same with those mentioned in section 2.2.1.

• VOICE: For verbs, the feature is Active or Passive; for nouns, it is null.

• POSITION: The word's position corresponding to its predicate: Left, Right or Self.

• PRED: The lemma plus sense of the word.

• PRED_POS: The part-of-speech of the predicate.

- LEFTM_WORD, LEFTM_POS, RIGHTM_WORD, RIGHTM_POS: Leftmost and rightmost word and their part-of-speech of the word.
- POS_PATH: All part-of-speech from the word to its predicate, including Up, Down, Left and Right, eg. "NN ↑ VV ↓ CC ↓ VV".
- DEPREL_PATH: Dependency relations from the word to its predicate, eg. "COMP ↑ RELC ↑ COMP ↓ ".
- ANC_POS_PATH, ANC_DEPREL_PATH: Similar to POS_PATH and DEPREL_PATH, part-of-speech and dependency relations from the word to the common ancestor with its predicate.
- PATH_LEN: Count of passing words from the word to its predicate.
- FAMILY: Relationship between the word and its predicate, including Child, Parent, Descendant, Ancestor, Sibling, Self and Null.
- PRED_CHD_POS, PRED_CHD_DEPREL: Part-of-speech and dependency relations of all children of the word's predicate.

For different languages, some features mentioned above are invalid and should be removed, and some extended features could improve the performance of the classifier. In our system we mainly focus on Chinese, therefore, WORD and VOICE should be removed when processing Chinese data set. We also adopt some features proposed by (Xue, 2008):

- POS_PATH_BA, POS_PATH_SB, POS_PATH_LB: BA and BEI are functional words that impact the order of arguments. In PropBank, BA words have the POS tag BA, and BEI words have two POS tags: SB (short BEI) and LB (long BEI).

3 Experimental Results

Our experiments are based on a PC with a Intel Core 2 Duo 2.1G CPU and 2G memory. Training and evaluation data (Taulé et al., 2008; Xue et al., 2008; Hajič et al., 2006; Palmer et al., 2002; Burchardt et al., 2006; Kawahara et al., 2002) have been converted to a uniform CoNLL Shared Task format. In all experiments, SVM and ME model are trained using training data, and tested with development data of all languages.

The system for closed challenge is designed as two parts. For syntactic dependency training and parsing, we utilize the projective model in Malt-Parser for data sets. We also follow default settings

in MaltParser, such as assigned parameters for LIBSVM and combined prediction strategy, and utilize improved approaches mentioned in section 2. For semantic dependency training and parsing, we choose the count of iteration as 100 and cutoff value as 10 for the ME model. Table 1 shows the training time for syntactic and semantic dependency of all languages. Parsing time for syntactic is not more than 30 minutes, and for semantic is not more than 5 minutes of each language.

	syn	prd	sem
English	7h	12min	47min
Chinese	8h	18min	61min
Japanese	7h	14min	46min
Czech	13h	46min	77min
German	6h	16min	54min
Spanish	6h	15min	55min
Catalan	6h	15min	50min

Table 1. Training cost for all languages. syn, prd and sem mean training time for syntactic dependency, predicate identification and semantic dependency.

3.1 Syntactic Dependency Parsing

We utilize MaltParser with improved algorithms mentioned in section 2.1 for syntactic dependency parsing, and the results are shown in Table 2.

	LAS	UAS	label-acc.
English	87.57	89.98	92.19
Chinese	79.17	81.22	85.94
Japanese	91.47	92.57	97.28
Czech	57.30	75.66	65.39
German	76.63	80.31	85.97
Spanish	76.11	84.40	84.69
Catalan	77.84	86.41	85.78

Table 2. Performance of syntactic dependency parsing

Table 2 indicates that parsing for Japanese and English data sets has a better performance than other languages, partly because determinative algorithm and history-based grammar are more suited for these two languages. To compare the performance of our approach of improved deterministic algorithm and extended features, we make another experiment that utilize original arc-standard algorithm and base features for syntactic experiments. Due to time limitation, the experiments are only based on Chinese training and evaluation data. The results show that LAS and UAS drops about 2.7% and 2.2% for arc-standard algorithm, 1.6% and 1.2% for base features. They indicate that our de-

terministic algorithm and the extend features can help to improve syntactic dependency parsing. We also notice that the results of Czech achieve a lower performance than other languages. It mainly because the language has more rich morphology, usually accompanied by more flexible word order. Although using a large training set, linguistic properties greatly influence the parsing result. In addition, extended features are not suited for this language and the feature model should be optimized individually.

For all of the experiments we mainly focus on the language of Chinese. When parsing Chinese data sets we find that the focus words where most of the errors occur are almost punctuations, such as commas and full stops. Apart from errors of punctuations, most errors occur on prepositions such as the Chinese word 'at'. Most of these problems come from assigning the incorrect dependencies, and the reason is that the parsing algorithm concerns the form rather than the function of these words. In addition, the prediction of dependency relation ROOT achieves lower precision and recall than others, indicating that MaltParser overpredicts dependencies to the root.

3.2 Semantic Dependency Parsing

MaxEnt is employed as our classifier to train and parse semantic dependencies, and the results are shown in Table 3, in which all criterions are labeled.

	P	R	F1
English	76.57	60.45	67.56
Chinese	75.45	69.92	72.58
Japanese	91.93	43.15	58.73
Czech	68.83	57.78	62.82
German	62.96	47.75	54.31
Spanish	40.11	39.50	39.80
Catalan	41.34	40.66	41.00

Table 3. Performance of semantic dependency parsing

As shown in Table 3, the scores of the latter five languages are quite lower than those of the former two languages, and the main reason could be inferred from the scores of Table 2 that the drop of the performance of semantic dependency parsing comes from the low performance of syntactic dependency parsing. Another reason is that, morphological features are not be utilized in the classifier. Our post experiments after submission show that average performance could improve the performance after adding morphological and some combined features. In addition, difference between precision and recall indicates that the classification procedure works better than the identification procedure in semantic role labeling.

For Chinese, semantic role of some words with part-of-speech VE have been mislabeled. It's mainly because that these words in Chinese have multiple part-of-speech. The errors of POS and PRED greatly influence the system to perform these words. Another main problem occurs on the pairs NN + A0/A1. Identification of the two pairs are much lower than VA/VC/VE/VV + A0/A1 pairs. The reason is that the identification of nominal predicates have more errors than that of verbal predicates due to the combination of SRL for these two kinds of predicates. For further study, verbal predicates and nominal predicates should be handled respectively so that the overall performance can be improved.

3.3 Overall Performance

The average performance of our system for all languages achieves 67.81% of macro F1 Score, 78.01% of syntactic accuracy, 56.69% of semantic labeled F1, 71.66% of macro precision and 64.66% of micro recall.

4 Conclusion

In this paper, we propose a pipelined approach for CoNLL-09 shared task on joint learning of syntactic and semantic dependencies, and describe our system that can handle multiple languages. Our system focuses on improving the performance of syntactic and semantic dependency respectively. Experimental results show that the overall performance can be improved for multiple languages by long distance dependency algorithm and extended history-based features. Besides, the system fits for verbal predicates than nominal predicates and the classification procedure works better than identification procedure in semantic role labeling. For further study, respective process should be handled between these two kinds of predicates, and argument identification should be improved by using more discriminative features for a better overall performance.

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. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006).* Genoa, Italy.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pp.340–345.

Kadri Hacioglu. 2004. Semantic Role Labeling Using Dependency Trees. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová and Zdeněk Žabokrtský. 2006. *The Prague Dependency Treebank 2.0.* CD-ROM. Linguistic Data Consortium, Philadelphia, Pennsylvania, USA. ISBN 1-58563-370-4. LDC Cat. No. LDC2006T01. URL: http://ldc.upenn.edu.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antonia 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. *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009).* Boulder, Colorado, USA. June 4-5. pp.3-22.

Daisuke Kawahara, Sadao Kurohashi and Koiti Hasida. 2002. Construction of a Japanese Relevance-tagged Corpus. *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002).* Las Palmas, Spain. pp.2008-2013.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.91–98.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL)*, pp.49–56.

Joakim Nivre. 2004. Incrementality in Deterministic Dependency Parsing. In Incremental Parsing: Bringing Engineering and Cognition Together. *Workshop at ACL-2004*, Barcelona, Spain, pp.50-57.

Joakim Nivre and Johan Hall. 2005. MaltParser: A language-independent system for data-driven dependency parsing. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT).*

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kubler, Svetoslav Marinov and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural language Engineering*, Volume 13, Issue 02, pp.95-135.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin and Daniel Jurafsky. 2005. Support Vector Learning for Semantic Argument classification. *Machine Learning Journal*, 2005, 60(3): 11−39.

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 *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008).*

Mariona Taulé, Maria Antònia Martí and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008).* Marrakech, Morocco.

Liu Ting, Wanxiang Che, Sheng Li, Yuxuan Hu, and Huaijun Liu. 2005. Semantic role labeling system using maximum entropy classifier. In *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL).*

Nianwen Xue. 2008. Labeling Chinese Predicates with Semantic roles. *Computational Linguistics*, 34(2): 225-255.

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

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pp.195–206.