

An Iterative Approach for Joint Dependency Parsing and Semantic Role Labeling

Qifeng Dai

Department of Computer Science,
University of Science and
Technology of China, Hefei,
China
daiqifeng001@126.com

Enhong Chen

Department of Computer Science,
University of Science and
Technology of China, Hefei,
China
cheneh@ustc.edu.cn

Liu Shi

Department of Computer Science,
University of Science and
Technology of China, Hefei,
China
shiliu@ustc.edu

Abstract

We propose a system to carry out the joint parsing of syntactic and semantic dependencies in multiple languages for our participation in the shared task of CoNLL-2009. We present an iterative approach for dependency parsing and semantic role labeling. We have participated in the closed challenge, and our system achieves 73.98% on labeled macro F1 for the complete problem, 77.11% on labeled attachment score for syntactic dependencies, and 70.78% on labeled F1 for semantic dependencies. The current experimental results show that our method effectively improves system performance.

1 Introduction

In this paper we describe the system submitted to the closed challenge of the CoNLL-2009 shared task on joint parsing of syntactic and semantic dependencies in multiple languages.

Given a sentence, the task of dependency parsing is to identify the syntactic head of each word in the sentence and classify the relation between the dependent and its head. The task of semantic role labeling is to label the senses of predicates in the sentence and labeling the semantic role of each word in the sentence relative to each predicate.

The difficulty of this shared task is to perform joint task on dependency parsing and semantic role labeling. We split the shared task into four sub-problems: syntactic dependency parsing, syntactic dependency label classification, word sense disambiguation, and semantic role labeling. And we pro-

pose a novel iterative approach to perform the joint task. In the first step, the system performs dependency parsing and semantic role labeling in a pipelined manner and the four sub-problems extract features based on the known information. In the iterative step, the system performs the four tasks in a pipelined manner but uses features extracted from the previous parsing result.

The remainder of the paper is structured as follows. Section 2 presents the technical details of our system. Section 3 presents experimental results and the performance analysis. Section 4 looks into a few issues concerning our forthcoming work for this shared task, and concludes the paper.

2 System description

This section briefly describes the main components of our system: a) system flow; b) syntactic parsing; c) semantic role labeling; d) an iterative approach to perform joint syntactic-semantic parsing.

2.1 System flow

As many systems did in CoNLL Shared Task 2008, the most direct way for such task is pipeline approach. First, Split the system into four subtasks: syntactic dependency parsing, syntactic dependency relation labeling, predicate sense labeling and semantic role labeling. Then, execute them one by one. In our system, we extend this pipeline system to an iterative system so that it can do a joint labeling to improve the performance.

Our iterative system is based on the pipeline system. For the first iteration (original step), we use the pipeline system to parse and label the

whole sentence. For the rest iterations (iterative step), we use another pipeline system to parse and label it. The structure of this pipeline is the same as the original one, but each subtask can have much more features than the original subtask. Because the whole sentence has been labeled in the original step, all information is available for every subtask. For example, when doing syntactic dependency relation labeling, we can add some features about sense and semantic role. It seems like using syntactic results to do semantic labeling, then using semantic results to improve syntactic labeling. This is the core idea of our joint system. Figure 1 shows the main flow of our system.

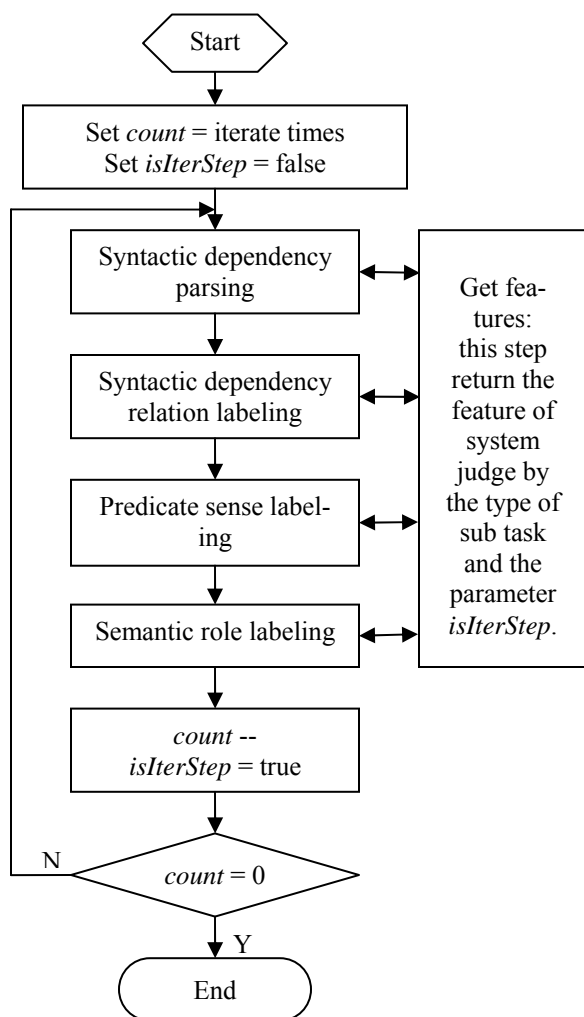


Figure 1. The main flow of iteration system

2.2 Dependency Parsing

In the dependency parsing step, we split the task into two sub-problems: syntactic dependency parsing and syntactic dependency relation labeling.

In the syntactic dependency parsing stage, MSTParser¹, a dependency parser that searches for maximum spanning trees over directed graphs, is applied. Due to the differences between the seven languages, we use different parameters to train a parsing model. Specifically, as Czech and German languages are none-projective and the others are projective, we train Czech and German languages with parameter “none-projective” and the others with “projective”.

On the syntactic dependency label classification step, we used the max-entropy classification algorithm to train the model. This step contains two processes. In the first process the sub-problem trains the model with the following basic features:

- FORM1: FORM of the head.
- LEMMA1: LEMMA of the head.
- STEM1 (English only): STEM of the head.
- POS1: POS of the head.
- IS_PRED1: the value of FILLPRED of the head.
- FEAT1: FEAT of the head.
- LM_STEM1 (English only): the left-most modifier’s STEM of head.
- LM_POS1: the left-most modifier’s POS of head.
- L_NUM1: number of the head’s left modifiers.
- RM_STEM1 (English only): the right-most modifier’s STEM of head.
- RM_POS1: the right-most modifier’s POS of head.
- M_NUM1: number of modifiers of the head.
- SUFFIX1 (English only): suffix of the head.
- FORM2: FORM of the dependent.
- LEMMA2: LEMMA of the dependent.
- STEM2 (English only): STEM of the dependent.
- POS2: POS of the dependent.
- IS_PRED2: the value of FILLPRED of the dependent.

¹ <http://sourceforge.net/projects/mstparser>

- FEAT2: FEAT of the dependent.
- LM_STEM2 (English only): the left-most modifier's STEM of dependent.
- LM_POS2: the left-most modifier's POS of dependent.
- L_NUM2: number of the dependent's left modifiers.
- RM_STEM2 (English only): the right-most modifier's STEM of dependent.
- RM_POS2: the right-most modifier's POS of dependent.
- M_NUM2: number of modifiers of the dependent.
- SUFFIX2 (English only): suffix of the dependent.
- DEP_PATH_ROOT_POS2: POS list from dependent to tree's root through the syntactic dependency path.
- DEP_PATH_ROOT_LEN2: length from dependent to tree's root through the syntactic dependency path.
- POSITION: The position of the word with respect to its predicate. It has three values, "before", "is" and "after", for the predicate.

In the iterative step, in addition to the features mentioned above, the sub-task trains the model with the following features:

- DEP_PATH_ROOT_POS1: POS list from head to tree's root through the syntactic dependency path.
- DEP_PATH_ROOT_REL1: length from dependent to tree's root through the syntactic dependency path.
- PRED_POS: POS list of all predicates in the sentence.
- FORM2 + DEP_PATH_REL: component of FORM2 and the POS list from head to the dependent through the syntactic dependency path.
- POSITION + FORM2
- STEM1 + FORM2 (English only)
- STEM1 + STEM2 (English only)
- POSITION + POS2
- ROLE_LIST2: list of APRED when the dependent is a predicate.
- ROLE: list of APRED and PRED when the head is predicate.
- L_ROLE: the nearest semantic role in its left side when head is a predicate.

- R_ROLE: the nearest semantic role in its right side when head is a predicate.
- IS_ROLE1: whether dependent is a semantic role of head when head is a predicate.

2.3 Semantic role labeling

Unlike CoNLL-2008 shared task, this shared task does not need to identify predicates. So the main task of this step is to label the sense of each predicate and label the semantic role for each predicate.

When labeling the sense of each predicate, we build a classification model for each predicate. As the senses of different predicates are usually unrelated even if they have the same sense label, this makes it difficult for us to use only one classifier to label them. But this approach leads to another issue. The set of predicates in the training set cannot cover all predicates. For new predicates in the test set, no classification model can be found for them, and we build a most common sense for them. The features we used are as follow:

- DEPREL1: DEPREL of the predicate.
- STEM1
- POS1
- RM_STEM1 (English only)
- RM_POS1
- FORM2
- POS2
- SUFFIX2
- VOICE (English only): VOICE of predicate.
- POSITION + POS2
- L_POS1 + POS1 + R_POS1: component of left word's POS and predicate POS and right word's POS.
- FORM2 + DEP_PATH_REL
- DEP_PATH_ROOT_POS1
- DEP_PATH_ROOT_REL1

When labeling the semantic role, we use a similar approach as we did in CoNLL Shared Task 2008. However, as the frames information is not supplied for all languages, we do not use it in this task. The features we use are as follows:

- DEPREL1
- STEM1 (English only)
- POS1
- RM_STEM1 (English only)
- RM_POS1

- FORM2
- POS2
- SUFFIX2
- VOICE2 (English only)
- POSITION
- DEP_PATH_REL
- DEP_PATH_POS
- SENSE2
- SENSE2 + VOICE2
- POSITION + VOICE2
- DEP_PATH_LEN
- DEP_PATH_ROOT_REL1

Moreover, we build an iterative model in this shared task. When doing an iterative labeling, the previous labeling results are known. So we can design some new features for checking the previous results in a global view. The features we add for the iterative model are as follows:

- SENSE1: SENSE of the predicate.
- SENSE1 + VOICE1: component of the SENSE + VOICE of predicate.
- VOICE1 + FORM1: component of VOICE and FORM.
- ROLE_LIST1: list of APRED of predicate.

2.4 Iterative Approach

As described above, some subtasks have two groups of features. One is for the pipeline model, and the other is for the iterative model. The usage of these two types of model is the same. The only difference is that they use different features. The iterative model can get more information, so they can use more features. These additional features can contain some joint and global (like frame and global structure) information. The performance may be improved because the viewer is extended. Some structural error and semantic conflict can be fixed.

Although the usage of the two types of model is the same, there are some differences when building the models.

In the iterative step, all information is available for doing parsing and labeling. For example, when doing syntactic dependency relation labeling in the iterative step, the fields “HEAD”, “DEPREL”, “PRED” and “APREDS” are filled by the pervious iteration. So all these information can be used in the iterative step. This will cause one issue: use “HEAD1” to label “HEAD2”. When training the

model, “HEAD1” is golden. The classifier will build a model directly and let “HEAD2” equal to “HEAD1”. However, in the iterative step, “HEAD1” is not golden, but such model makes it impossible to change the results.. The iterative step will be useless.

We design a simple method to avoid this issue.

- Firstly, split the training set into N (N>1) subsets.
- Secondly, for each subset, use the left N-1 subsets to build an original sub-model (use features in the pipeline step).
- Thirdly, use each sub-model to label the corresponding subset.
- Lastly, use these labeled N subsets to extract samples (use features in the iterative step) for building the iterative model.

In this way, the “HEAD1” is not golden any more. And for each sub-task, we can use the similar method to build the original model and the iterative model.

Moreover, in our system, we only build the iterative models for syntactic dependency relation labeling and semantic role labeling. For syntactic dependency parsing, we use an approach with very high time and space complexity, so it is not added to the iterative step. Thus, its results will not be changed in the iterative step. For sense labeling, we build classification models for every predicate. There are too many models and each model contains only a few classes. We think they are not suitable for building the iterative model. But, as its previous sub-task (syntactic dependency relation labeling) is added to the iterative step, it is useful to add it to the iterative step. Though we do not build an iterative model for sense labeling, we can directly use its pipeline model. This is another advantage of our iterative model: if one subtask is not suitable for doing iterative labeling/parsing, we can use its pipeline model instead.

3 Experiments and Results

We have tested our system with the test set and obtained official results as shown in Table 1. We have tried to find how the iterative step influences syntactic dependency parsing and semantic role labeling. For syntactic dependency parsing and semantic role labeling, we do experiments on the test set.

	Macro F1 Score
Average	73.98
Catalan	72.09
Chinese	72.72
Czech	67.14
English	81.89
German	75.00
Japanese	80.89
Spanish	68.14

Table 1. The Macro F1 Score of every languages and the average value.

3.1 Syntactic Dependency Parsing

Dependency Parsing can be split into two sub-problems: syntactic dependency parsing and syntactic dependency label classification. We use the iterative method on syntactic dependency label classification. We do experiments on the test set.

On the test set, we do two group experiments. In the first group, we build a subtest to test this sub-task only. All other information is given, and we just label the dependency relation. The results are shown in Table 2. The row of “Initial step” shows the results of this sub task in the original step. The left two rows show the results in the iterative step with iterating once and twice. The table shows that the iterative approach improves the performance. Especially for Catalan, the performance increases by 2.89%.

Certainly, in the whole system, this subtask cannot get golden information about sense and semantic roles. So we test it in the whole system (joint test) on the test set in the second group of experiments. As shown in Table 3, the iterative step is not as good as previous test. But it is still useful for some languages. The reason that some languages have no improvements on the iterative step is that the result of the initial step is not so good.

3.2 Semantic Role Labeling

Like syntactic dependency parsing, we do two tests on Semantic Role Labeling. This result is not consistent with the official data because we have added some features of the subtask. The results of subtest can be found in Table 4. And Table 5 shows the results of the joint test. These two groups of results show that the advantage of the iterative step is not as good as that of syntactic dependency labeling in subtest. But it improves the performance for most languages. The iterative step improves the performance in both two tests.

3.3 Analysis of Results

From the experimental results, we can see that the effect of each part of the iterative step depends on the overall labeling result of the previous step. And the labeling effect varies with different languages. Iterative approach can improve the performance of the system but it strongly depends on the initial labeling result.

4 Conclusion and Future Work

This paper has presented a simple discriminative system submitted to the CoNLL-2009 shared task to address the learning task of syntactic and semantic dependencies. The paper first describes how to carry out syntactic dependency parsing and semantic role labeling, and then a new iterative approach is presented for joint parsing. The experimental results show that the iterative process can improve the labeling accuracy on syntactic and semantic analysis. However, this approach probably depends on the accuracy of the initial labeling results. The results of the initial labeling results will affect the effect of the iterative process.

Because of time constraints and inadequate experimental environment, our first results do not meet our expectation, and the effect of the iterative step is not so clear. Next, we will strive to refine our approach to produce good results for the syntactic dependency parsing, since it has a great impact on the final parsing results.

Acknowledgments

The authors would like to thank the reviewers for their helpful comments. This work was supported by National Natural Science Foundation of China (No.60573077, No.60775037) and the National High Technology Research and Development Program of China (863 Program) (grant no. 2009AA01Z123). We also thank the High-Performance Center of USTC for providing us with the experimental platform.

	Average	Catalan	Chinese	Czech	Czech-ood	English	English-ood	German	German-ood	Japanese	Spanish
Initial step	93.64	95.66	95.01	88.10	88.10	96.79	92.98	96.41	89.71*	98.17	95.48
Iteration 1	94.60	98.56*	96.08*	88.59	88.29	97.31*	94.57*	96.63*	89.31	98.34	98.30
Iteration 2	94.65	98.55	96.08*	88.68*	88.45*	97.29	94.56	96.63*	89.53	98.35*	98.33*

Table 2. The subtest result of Labeled Syntactic Accuracy of each language and the average performance value on test set. (* denotes the best score for the system)

	Average	Catalan	Chinese	Czech	Czech-ood	English	English-ood	German	German-ood	Japanese	Spanish
Initial step	74.02	77.75	73.81	58.69*	55.50*	84.75	78.85	82.45	66.27*	90.45*	71.64
Iteration 1	73.90	77.82	73.86*	58.17	54.95	84.81	78.95	82.51*	65.78	90.43	71.68
Iteration 2	73.94	77.85*	73.86*	58.31	55.13	84.82*	79.02*	82.46	65.85	90.45*	71.69*

Table 3. The joint test result of Labeled Syntactic Accuracy of each language and the average performance value on test set. (* denotes the best score for the system)

	Average	Catalan	Chinese	Czech	Czech-ood	English	English-ood	German	German-ood	Japanese	Spanish
Initial step	83.83	88.56	85.86	88.08	86.20*	86.23	82.09	80.98	78.82	74.32*	87.45
Iteration 1	84.34	89.02*	87.14*	87.88	86.09	86.66	82.07	83.66*	79.28*	74.06	87.59
Iteration 2	84.36	89.02*	87.01	88.10*	86.17	86.78*	82.34*	83.15	79.18	74.06	87.81*

Table 4. The sub test result of Semantic Labeled F1 of each language and the average performance value on test set. (* denotes the best score for the system)

	Average	Catalan	Chinese	Czech	Czech-ood	English	English-ood	German	German-ood	Japanese	Spanish
Initial step	70.01	66.87	71.63	75.50	75.71	78.97	69.87	67.50	58.47	70.91*	64.64
Iteration 1	70.15	67.12	71.98	75.54	75.68	79.40	70.17*	68.08*	58.55*	70.69	64.32
Iteration 2	70.20	67.33*	71.99*	75.65*	75.90*	79.47*	69.98	67.98	58.33	70.70	64.65*

Table 5. The joint test result of Semantic Labeled F1 of each language and the average performance value on test set. (* denotes the best score for the system)

References

- 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.
- Mariona Taulé, Maria Antònia Martí and Marta Recasens. 2008. AnCorra: Multilevel Annotated Corpora for Catalan and Spanish. Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008). Marrakech, Morocco.
- Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the Chinese Treebank. *Natural Language Engineering*, 15(1):143-172.
- Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Míková 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>.
- Surdeanu, Mihai, 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).
- 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.
- 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.
- McDonald, Ryan. 2006. Discriminative learning and Spanning Tree Algorithms for Dependency parsing. Ph.D. thesis, University of Pennsylvania.
- Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical. *Report CMU-CS-99-108*.