Probabilistic Model for Syntactic and Semantic Dependency Parsing

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

Dawei Hu
Department of Computer Science, University of Science and Technology of China, Hefei, China
dwhu@mail.ustc.edu.cn

Abstract

This paper proposes a novel method to analyze syntactic dependencies and label semantic dependencies around both the verbal predicates and the nouns. In this method, a probabilistic model is designed to obtain a global optimal result. Moreover, a predicate identification model and a disambiguation model are proposed to label predicates and their senses. The experimental results obtained on the wsj and brown test sets show that our system obtains 77% of labeled macro F1 score for the whole task, 84.47% of labeled attachment score for syntactic dependency task, and 69.45% of labeled F1 score for semantic dependency task.

1 Introduction

There are two difficulties in the CoNLL 2008 shared task. One is how to label semantic role on a dependency-based representation and how to label verbal predicates and nouns. The other one is how to combine the syntactic task with the semantic task together.

On the basis of statistical analysis of labeling results, we optimize the traditional approaches of syntactic dependency parsing and semantic role labeling. Moreover, we design a predicate identification model and a disambiguation model, which will be described in section 2.3, for labeling predicates and their senses. In the disambiguation model, an exhaustion method is used to find the best sense which is corresponding to a frame of predicate. In order to obtain a global optimization result for every sentence, a probabilistic model is designed to combine all subtasks.

The rest of this paper is organized as follows: our system is described in section 2; and section 3 reports our results on development and test sets; at last we conclude the paper in section 4.

2 A Probabilistic Model for Syntactic and Semantic Dependency Labeling

Compared with previous tasks, this shared task is more complex. It aims to merge both syntactic and semantic dependencies under a unified representation. Obviously, it can be divided into two subtasks: syntactic dependency parsing and semantic dependency labeling. For the second subtask, predicates and their senses should be labeled before semantic arguments for predicates are labeled. Since many predicates have only one sense, it is inefficient to build a multi-label classifier to classify each predicate. When a classification approach is used, it is mandatory to consider multiple senses for those predicates with only one or two senses. To prevent assigning irrelevant senses to predicates, we do not adopt classification approach. Instead, two more subtasks, i.e., predicate identification and predicate sense labeling, are introduced in this paper. The predicate sense labeling and semantic dependency labeling are performed together with a disambiguation model.

To ensure that we can get an optimal overall syntactic and semantic dependency results through integrating the above steps, a probability model is proposed. The probabilistic model is described in Equation (1), where the score P_{sent} of a sentence labeling is the combined conditional probability of its all subtasks, P_{syn} is the probability of syntactic dependency parsing, P_{pred} is the probability of predicate identification, $P_{sem}(i)$ is the probability of

CoNLL 2008: Proceedings of the 12th Conference on Computational Natural Language Learning, pages 263–267
Manchester, August 2008

semantic dependency labeling for the *ith*-predicate, and *n* is the number of predicates.

$$P_{\text{sent}} = P_{\text{syn}} * P_{\text{pred}} * \prod_{\text{i}=1}^{\text{n}} P_{\text{sem}}(i) \qquad (1)$$

For each sentence, its top-*N* candidates using syntactic dependency parsing are obtained. Then for each candidate, predicates and semantic arguments are labeled. At last, the best one with the highest P_{sent} is chosen as final labeling result.

2.1 Syntactic Dependency Parsing

There are several approaches for syntactic dependency parsing, as demonstrated in the CoNLL 2007 shared task. A commonly used LR algorithm is applied to this task. Unlike the best-first probabilistic shift-reduce LR algorithm used by (Kenji and Jun, 2007), here a combined probability of all parsing steps is used to evaluate parsing results, and the best one is obtained as the final result. The probability of syntactic dependency parsing is defined in Equation (2).

$$P_{syn} = \prod_{i=1}^{j} P_{act}(i) \qquad (2)$$

where $P_{act}(i)$ is the probability of every LR action *act* at step *i*, *a*nd *j* is the number of all steps.

As the search space of LR parser is exponential growth with the word number, the maximum size of candidate states is limited to 50.

The features that we use are similar to (Kenji and Jun, 2007). Hence we do not describe them in this paper.

2.2 Predicate Identification

In this subtask, a MaxEnt model is adopted for classification. The features we used are as follow:

- **Base info:** FORM, LEMMA, POS (GPOS if available, or is PPOS), SPLIT_FORM, SPLIT_LEMMA, PPOSS.
- **Base syntactic dependency info:**
  - Number of modifiers;
  - Number of modifiers of the previous word;
  - Number of modifiers of the next word;
  - PPOSS of left-most modifier;
  - Deprel of left-most modifier;
  - PPOSS of right-most modifier;
  - Deprel of right-most modifier.
- **Modifiers info**
  - POS list of all modifiers: if GPOS is available, POS is GPOS. Otherwise it is PPOS.
  - DEPREL list of all modifiers;
  - SPLIT_LEMMA list of all modifiers;
  - PPOSS list of all modifiers.
- **Head's base info**
- **Head's base syntactic dependency info**
- **Head's modifiers info**
- **Deprel:** the syntactic dependency relation to head.
- **Word stem**
- **Stem of right-most modifier**
- **PPOSS of right-most modifier**
- **Suffix:** The suffix of the word. We use the last 3 characters as this feature.
- **Voice:** Check if the word is a verb and is passive voice.
- **Previous word info:** Check if the previous word is a predicate.
- **Pos path to ROOT:** PPOSS list from word to ROOT through the syntactic dependency path.
- **Deprel path to ROOT:** DEPREL list from word to ROOT through the syntactic dependency path.

Through statistical analysis, we find that PPOSS of nearly all predicates are in a particular category which contains NN, NNP, NNS, VB, VBD, VBG, VBN, VBP, VBZ, and JJ. Hence we ignore the words without these PPOSS to reduce the number of samples and speed up the process of training and recognition. Meanwhile, we also ignore the words having no relational frame in PropBank or NomBank.

2.3 Predicate Sense Labeling

In this subtask, we label the sense of each predicate. Different predicates are usually unrelated even if they have the same sense number, which makes us hardly use a classifier to label them. Hence, we design a disambiguation model to solve this problem.

Firstly, for each word which has been identified to be a predicate, we find out all of its probable sense forms (corresponding to the field of "PRED"). According to statistical analysis, only about 0.05% PREDs are not described in PropBank frames or NomBank frames. So it is reasonable to assume that all PREDs could be found in PropBank or NomBank. Moreover, we find that about 96% PREDs are formed as "SPLIT_LEMMA + .sense" or "LEMMA + .sense". As a result, when a word is identified to be a predicate, we use its LEMMA and SPLIT_LEMMA to find all possible PREDs from PropBank and NomBank. Furthermore, if some special words are unsuitable for these two forms, we should convert them into their original forms first and then find their possible PREDs.

For the rest anomalistic words, we build a mapping dictionary from training data.

Secondly, for each possible sense form, we label semantic argument for all words. If a word is not a semantic argument, it would be labeled as "_". The score of the current possible sense form is calculated as the combination of all probability of each labeling. More details about semantic dependency labeling will be described in section 2.4.

Thirdly, we choose the sense form and its semantic arguments with the highest score. The above steps will be repeated until all predicates have definite senses.

2.4 Semantic Dependency Labeling

Unlike CoNLL-2005 shared task, this shared task performing Semantic Role Labeling on a dependency-based representation (DSRL). It is a novel way for SRL and the traditional SRL methods can not directly be used here. Constituent-based SRL model needs to find out all probable constituents, while DSRL only considers the semantic dependency between word and predicate. Moreover, DSRL uses syntactic dependency parsing tree instead of traditional full syntactic parsing tree. As a result, the traditional features need to be amended accordingly. The features we used are as follows:

- **Deprel**
- **Word stem**
- **POS:** if GPOS is available, POS is GPOS. Otherwise it is PPOS.
- **Stem of right-most modifier**
- **PPOSS of right-most modifier**
- **Predicate:** the FORM of predicate.
- **PPOSS of predicate**
- **Suffix of predicate**
- **Voice:** voice of predicate
- **Position:** The position of the word with respect to its predicate. It has three values, "before", "is" and "after", for the predicate.
- **Deprel path to predicate:** DEPREL list from word to its predicate through the syntactic dependency path.
- **Length of syntactic dependency path to predicate**
- **Sense:** the sense of predicate

Moreover, we try to find more features with frames. Since the PropBank and NomBank are available and all predicates with senses are available for this subtask. Statistical analysis shows that nearly all core semantic arguments (AA, A0, A1, A2 …) of a predicate are described in the frame of predicate. But it is incorrect contrarily. Based on these observations, we design features the following features for five frequently used core arguments:

- **A0 is in predicate's frame:** Have two values: "YES" and "NO".
- **A1 is in predicate's frame**
- **A2 is in predicate's frame**
- **A3 is in predicate's frame**
- **A4 is in predicate's frame**

Because the other core semantic arguments are rare, we do not need to design features for them. With this method, the labeling efficiency is improved while the precision almost keeps unchanged.

As the frame information has been used in features, we do not add any valency check on the labeling result.

3 Experiments and Analysis

3.1 Data and Environment

The data provided for this Closed Challenge of shared task is part of TreeBank and Brown corpus. Training set covers sections 02-21 of TreeBank. Development set covers section 24 of TreeBank. Wsj test set covers section 23 of TreeBank. Brown test set covers sections ck01, ck02, and ck03 of the Brown corpus.

The maximum entropy classier (Berger et al, 1996) used is Le Zhang's Maximum Entropy Modeling Toolkit and the L-BFGS parameter estimation algorithm with gaussian prior smoothing (Chen and Rosenfeld, 1999). The gaussian prior is set to 2 and the iteration count is set to 500. All results we list here are post-evaluated because there are some small modifications.

The experiments are performed on a PC with AMD Athlon™ 64 x2 4400+ CPU and 2GB main memory running Microsoft Windows XP with sp2. Our system is developed using C++.

In our experimental analysis, the abbreviations used are listed as follows:

- LAS_1: Labeled attachment score
- UAS: Unlabeled attachment score
- LAS_2: Label accuracy score
- LP: Labeled precision
- LR: Labeled recall
- LF1: Labeled F1
- UP: Unlabeled precision
- UR: Unlabeled recall
- UF1: Unlabeled F1

3.2 Syntactic Dependency Parsing

We trained two LR models for syntactic dependency parsing. The first LR model uses MaxEnt classification to determine possible parser actions and their probabilities. The second LR model also uses MaxEnt classification, but parsing is performed backwards simply by reversing the sentence before parsing starts.

For a sentence, each model can label top-*N* candidates and calculate the probability for every result. We join these two models by finding the candidate with the highest probability from all candidates as the final result for the sentence. Table 1 shows the results of each model and joint model. We can see that the two LR models obtain similar results. The joint model can obtain better result and increase almost one percentage. The processing time of joint model is twice more than that of the two other models.

		LR Model	LR-back Model	Joint Model
dev	LAS_1	83.05	83.38	84.43
	UAS	86.36	86.74	87.74
	LAS_2	89.15	89.63	90.08
wsj	LAS_1	84.84	84.06	85.48
	UAS	87.60	86.74	88.13
	LAS_2	90.70	90.47	91.21
brown	LAS_1	77.29	76.95	78.91
	UAS	82.75	82.61	84.38
	LAS_2	85.00	84.82	85.76
wsj + brown	LAS_1	84.00	83.27	84.75
	UAS	87.06	86.28	87.71
	LAS_2	90.07	89.84	90.6
Speed (sec/sent)		0.49	0.42	0.92

Table 1: Syntactic dependency parsing results

3.3 Predicate Identification

Our predicate identification approach is described in section 2.2. We use the gold HEAD and DEPREL fields to test our approach. The results are shown in Table 2. The labeling for each sentence spends about 14ms.

	dev	wsj	brown
Precision	93.56	93.61	87.51
Recall	93.24	93.39	89.04
F1	93.40	93.50	88.27

Table 2: Predicate identification results

3.4 Semantic Dependency Labeling

Semantic dependency labeling is the last subtask. Our DSRL model uses MaxEnt classification to determine the semantic dependency between each word and its corresponding predicate. The gold HEAD and DEPREL and PRED fields is used to test the model.

Statistical analysis shows that, for about 99% semantic argument labels, the length of syntactic dependency path from word to predicate is less than 7. So we ignore the words with the length of 7 or more.

The final results of semantic dependency labeling are shown in Table 3. The labeling for each sentence spends about 10ms.

Brown set is an out-of-domain set and wsj set is an in-domain set. Usually, the results on wsj are much better than those on brown. But here we found that the unlabeled scores are nearly the same between wsj and brown. It shows that our model performs well at unlabeled labeling on out-of-domain set, and should be improved at labeled labeling.

	dev	wsj	brown
LP	80.50	82.47	77.29
LR	70.73	73.58	67.16
LF1	75.30	**77.77**	**71.87**
UP	92.10	92.65	92.87
UR	80.92	82.65	80.69
UF1	86.15	**87.36**	**86.35**

Table 3: Semantic dependency labeling results

3.5 Overall Result

As described in section 2, we use a probabilistic model to integrate all subtasks. In the probabilistic model, syntactic dependency parsing should parse top-*N* candidate results. We do the rest parsing for each candidate result and get *N* integrated results. Then, for each integrated result, its P_{sent} is calculated and the best one is chose as the final result.

The DSRL results around verbal predicates and nouns on wsj set are shown in Table 4. It shows that verbal predicates are labeled much better than nouns.

	Unlabeled Predicate	Labeled Predicate	Labeled Semantic Arguments
NN*	87.79	79.52	58.09
VB*	96.85	80.25	73.77

Table 4: The F1 values of DSRL around verbal predicates and nouns on wsj

Table 5 shows the overall results with different *N*. The results are improved when *N* changes from 1 to 2. However, there is nearly no improvement by increasing *N* from 2 to 3. So *N* is set to be 2 in our system. Meanwhile, the effect of this approach is not obvious. We find that there are nearly only one or two different points between the top-2 candidate dependency parsing results. This leads to that the DSRL results with these top-2 candidate results are almost the same. This is the probable reason that the approach is not much improved with the increase of *N*. In the future it would be necessary for us to consider the number of different points when finding the top-*N* dependency results.

		N=1	N=2	N=3
dev	LP	78.58	78.93	79.01
	LR	75.58	75.52	75.33
	LF1	77.05	**77.19**	77.13
	UP	86.56	86.95	87.07
	UR	83.04	82.94	82.75
	UF1	84.76	84.90	84.85
wsj	LP	79.41	79.76	79.96
	LR	76.67	76.59	76.49
	LF1	78.02	78.15	78.19
	UP	86.59	86.92	87.11
	UR	83.40	83.25	83.10
	UF1	84.97	85.04	85.06
brown	LP	70.52	70.95	70.79
	LR	68	67.88	67.54
	LF1	69.24	69.38	69.13
	UP	81.87	82.39	82.28
	UR	78.65	78.47	78.14
	UF1	80.23	80.39	80.16
wsj + brown	LP	78.45	78.8	78.96
	LR	75.72	75.64	75.5
	LF1	77.06	**77.18**	77.19
	UP	86.08	86.43	86.59
	UR	82.89	82.73	82.56
	UF1	84.45	84.54	84.53
Speed (sec/sent)		0.93	0.94	0.95

Table 5: Overall macro scores (Wsem = 0.50)

4 Conclusion

We divide this shared task into four subtasks: syntactic dependency parsing, predicate identification, predicate sense labeling and semantic dependency labeling. Then, we design a probabilistic model to combine them. The purpose of our system is to find a global optimal result for every sentence. If a syntactic dependency parsing result has the highest probability but it is unreasonable, it would be difficult to get a semantic parsing result with high probability again. Hence, a more reasonable result may be found with lower syntactic dependency parsing probability.

In our system, we have not distinguished between nouns and verbal predicates. The experimental results show that the results of verbal predicates are much better than those of nouns. In the future, it is necessary for us to deal with them separately.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.60573077, No.60775037), Specialized Research Fund for the Doctoral Program of Higher Education (No.2007105), and Program for New Century Excellent Talents in University (No.NCET-05-0549).

References

Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to naturallanguage processing. *Computational Linguistics, 22(1):39–71.*

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

Gildea Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics, 28(3):245–288.*

Duan Xiangyu, Zhao Jun and Xu Bo. 2007. Probabilistic Parsing Action Models for Multi-Lingual Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007.*

Hacioglu K. 2004. Semantic Role Labeling Using Dependency Trees. In *Proceedings of COLING-2004.*

Johansson R. and Nugues P. 2007. Extended Constituent-to-dependency Conversion for English. In *Proceedings of NODALIDA 2007.*

Sagae, Kenji and Tsujii, Jun'ichi. 2007. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007.*

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. *Technical Report CMU-CS-99-108.*

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).*

Tsai Tzong-Han, Chia-Wei Wu, Yu-Chun Lin, and Wen-Lian Hsu. 2005. Exploiting full parsing information to label semantic roles using an ensemble of me and svm via integer linear programming. In *Proceedings of Computational Natural Language Learning (CoNLL-2005).*