The Integration of Dependency Relation Classification and Semantic Role Labeling Using Bilayer Maximum Entropy Markov Models

Weiwei Sun and **Hongzhan Li** and **Zhifang Sui**
Institute of Computational Linguistics
Peking University
{weiwsun, lihongzhan.pku}@gmail.com, szf@pku.edu.cn

Abstract

This paper describes a system to solve the joint learning of syntactic and semantic dependencies. An directed graphical model is put forward to integrate dependency relation classification and semantic role labeling. We present a bilayer directed graph to express probabilistic relationships between syntactic and semantic relations. Maximum Entropy Markov Models are implemented to estimate conditional probability distribution and to do inference. The submitted model yields 76.28% macro-average F1 performance, for the joint task, 85.75% syntactic dependencies LAS and 66.61% semantic dependencies F1.

1 Introduction

Dependency parsing and semantic role labeling are becoming important components in many kinds of NLP applications. 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 consists of analyzing the propositions expressed by some target predicates. The integration of syntactic and semantic parsing interests many researchers and some approaches has been proposed (Yi and Palmer, 2005; Ge and Mooney, 2005). CoNLL 2008 shared task proposes the merging of both syntactic dependencies and semantic dependencies under a unique unified representation (Surdeanu et al., 2008). We explore the integration problem and evaluate our approach using data provided on CoNLL 2008.

This paper explores the integration of dependency relation classification and semantic role labeling, using a directed graphical model that is also known as Bayesian Networks. The directed graph of our system can be seen as one chain of observations with two label layers: the observations are argument candidates; one layer's label set is syntactic dependency relations; the other's is semantic dependency relations. To estimate the probability distribution of each arc and do inference, we implement a Maximum Entropy Markov Model (McCallum et al., 2000). Specially, a logistic regression model is used to get the conditional probability of each arc; dynamic programming algorithm is applied to solve the "argmax" problem.

2 System Description

Our DP-SRL system consists of 5 stages:

1. dependency parsing;
2. predicate prediction;
3. syntactic dependency relation classification and semantic dependency relation identification;
4. semantic dependency relation classification;
5. semantic dependency relation inference.

2.1 Dependency Parsing

In dependency parsing stage, MSTParser[1] (McDonald et al., 2005), a dependency parser that searches for maximum spanning trees over directed graphs, is used. we use MSTParser's default

[1] http://www.seas.upenn.edu/ strctlrn/MSTParser/MSTParser.html

CoNLL 2008: Proceedings of the 12th Conference on Computational Natural Language Learning, pages 243–247
Manchester, August 2008

Lemma and its POS tag
Number of children
Sequential POS tags of children
Lemma and POS of Neighboring words
Lemma and POS of parent
Is the word in word list of NomBank
Is the word in word list of PropBank
Is POS of the word is VB* or NN*

Table 1: Features used to predict target predicates

parameters to train a parsing model. In the third stage of our system, dependency relations between argument candidates and target predicates are updated, if there are dependency between the candidates and the predicates.

2.2 Predicate Prediction

Different from CoNLL-2005 shared task, the target predicates are not given as input. Our system formulates the predicate predication problem as a two-class classification problem using maximum entropy classifier MaxEnt[2] (Berger et al., 1996). Table 1 lists features used. We use a empirical threshold to filter words: if the "being target" probability of a word is greater than 0.075, it is seen as a target predicate. This strategy achieves a 79.96% precision and a 98.62% recall.

2.3 Syntactic Dependency Relation Classification and Semantic Dependency Relation Identification

We integrate dependency parsing and semantic role labeling to some extent in this stage. Some dependency parsing systems prefer two-stage architecture: unlabeled parsing and dependency classification (Nivre et al., 2007). Previous semantic role labeling approaches also prefer two-stage architecture: argument identification and argument classification. Our system does syntactic relations classification and semantic relations identification at the same time. Specially, using a pruning algorithm, we collect a set of argument candidates; then we classify dependency relations between argument candidates and the predicates and predict whether a candidate is an argument. A directed graphical model is used to represent the relations between syntactic and semantic relations.

[2]http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Lemma, POS tag voice of predicates
POS pattern of predicate's children
Is the predicate from NomBank or PropBank
Predicate class. This information is extracted form frame file of each predicate.
Position: whether the candidate is before or after the predicate
Lemma and POS tag of the candidate
Lemma and POS of Neighboring words of the candidate
Lemma and POS of sibling words of the candidate
Length of the constituent headed by the candidate
Lemma and POS of the left and right most words of the constituent of the candidate
Punctuation before and after the candidate
POS path: the chain of POS from candidate to predicate
Single Character POS path: each POS in a path is clustered to a category defined by its first character
POS Pattern (string of POS tags) of all candidates
Single Character POS Pattern of all candidates

Table 2: Features used for semantic role labeling

2.4 Semantic Dependency Relation Classification

This stage assigns the final argument labels to the argument candidates supplied from the previous stage. A multi-class classifier is trained to classify the types of the arguments supplied by the previous stage. Table 2 lists the features used. It is clear that the general type of features used here is strongly based on previous work on the SRL task (Gildea and Jurafsky, 2002; Pradhan et al., 2005; Xue and Palmer, 2004). Different from CoNLL-2005, the sense of predicates should be labeled as a part of the task. Our system assigns *01* to all predicates. This is a harsh tactic since it do not take the linguistic meaning of the argument-structure into account.

2.5 Semantic Dependency Relation Inference

The purpose of inference stage is to incorporate some prior linguistic and structural knowledge, such as "each predicate takes at most one argument of each type." We use the inference process intro-

duced by (Punyakanok et al., 2004; Koomen et al., 2005). The process is modeled as an integer Linear Programming Problem (ILP). It takes the predicted probability over each type of the arguments as inputs, and takes the optimal solution that maximizes the linear sum of the probability subject to linguistic constraints as outputs. The constraints are a subset of constraints raised by Koomen et al. (2005) and encoded as following: 1) No overlapping or embedding arguments; 2) No duplicate argument classes for A0-A5; 3) If there is an R-arg argument, then there has to be an arg argument; 4) If there is a C-arg argument, there must be an arg argument; moreover, the C-arg argument must occur after arg; 5) Given the predicate, some argument types are illegal. The list of illegal argument types is extracted from framefile.

The ILP process can improve SRL performance on constituent-based parsing (Punyakanok et al., 2004). In our experiment, it also works on dependency-based parsing.

3 Bilayer Maximum Entropy Markov Models

3.1 Sequentialization

The sequentialization of a argument-structure is similiar to the pruning algorithm raised by (Xue and Palmer, 2004). Given a constituent-based parsing tree, the recursive pruning process starts from a target predicate. It first collects the siblings of the predicate; then it moves to the parent of the predicate, and collects the siblings of the parent. In addition, if a constituent is a prepositional phrase, its children are also collected.

Our system uses a similar pruning algorithm to filter out very unlikely argument candidates in a dependency-based parsing tree. Given a dependency parsing tree, the pruning process also starts from a target predicate. It first collects the dependents of the predicate; then it moves to the parent of the predicate, and collects all the dependents again. Note that, the predicate is also taken into account. If the target predicate is a verb, the process goes on recursively until it reaches the root. The process of a noun target ends when it sees a *PMOD*, *NMOD*, *SBJ* or *OBJ* dependency relation. If a preposition is returned as a candidate, its child is also collected. When the predicate is a verb, the set of constituents headed by survivors of our pruning algorithm is a superset of the set of survivors of the previous pruning algorithm on the corresponding constituent-based parsing tree. This pruning algorithm will recall 99.08% arguments of verbs, and the candidates are 3.75 times of the real arguments. If the stop relation such as *PMOD* of a noun is not taken into account, the recall is 97.67% and the candidates is 6.28 times of arguments. If the harsh stop condition is implemented, the recall is just 80.29%. Since the SRL performance of nouns is very low, the harsh pruning algorithm works better than the original one.

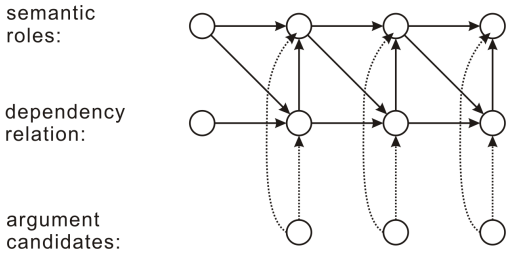

Figure 1: Directed graphical Model of The system

After pruning, our system sequentializes all argument candidates of the target predicate according to their linear order in the given sentence.

3.2 Graphical Model

Figure 1 is the directed graph of our system. There is a chain of candidates $\mathbf{x} = (x_0 = BOS, x_1, ..., x_n)$ in the graph which are observations. There are two tag layers in the graph: the up layer is information of semantic dependency relations; the down layer is information of syntactic dependency relations.

Given $\mathbf{x}$, denote the corresponding syntactic dependency relations $\mathbf{d} = (d_0 = BOS, d_1, ..., d_n)$ and the corresponding semantic dependency relations $\mathbf{s} = (s_0 = BOS, s_1, ..., s_n)$. Our system labels the syntactic and semantic relations according to the conditional probability in argmax flavor. Formally, labels the system assigned make the score $p(\mathbf{d}, \mathbf{s}|\mathbf{x})$ reaches its maximum. We decompose the probability $p(\mathbf{d}, \mathbf{s}|\mathbf{x})$ according to the directed graph modeled as following:

$$\begin{aligned} p(\mathbf{d}, \mathbf{s}|\mathbf{x}) &= p(s_1|s_0, d_1; \mathbf{x})p(d_1|s_0, d_0; \mathbf{x}) \cdots \\ &\quad p(s_{i+1}|s_i, d_{i+1}; \mathbf{x})p(d_{i+1}|s_i, d_i; \mathbf{x}) \cdots \\ &\quad p(s_n|s_{n-1}, d_n; \mathbf{x})p(d_n|s_{n-1}, d_{n-1}; \mathbf{x}) \\ &= \prod_{i=1}^{n} p(s_i|s_{i-1}, d_i; \mathbf{x})p(d_i|s_{i-1}, d_{i-1}; \mathbf{x}) \end{aligned}$$

Lemma, POS tag voice of predicates
POS pattern of predicate's children
Lemma and POS tag of the candidate
Lemma and POS of Neighboring words of the candidate
Lemma and POS of sibling words of the candidate
Length of the constituent headed by the candidate
Lemma and POS of the left and right most words of the constituent of the candidate
Conjunction of lemma of candidates and predicates; Conjunction of POS of candidates and predicates
POS Pattern of all candidates

Table 3: Features used to predict syntactic dependency parsing

3.3 Probability Estimation

The system defines the conditional probability $p(s_i|s_{i-1}, d_i; \mathbf{x})$ and $p(d_i|s_{i-1}, d_{i-1}; \mathbf{x})$ by using the maximum entropy (Berger et al., 1996) framework Denote the tag set of syntactic dependency relations $\mathcal{D}$ and the tag set of semantic dependency relations $\mathcal{S}$. Formally, given a feature map ϕ_s and a weight vector $\mathbf{w}_s$,

$$p_{\mathbf{w}_s}(s_i|s_{i-1}, d_i; \mathbf{x}) = \frac{\exp\{\mathbf{w}_s \cdot \phi_s(\mathbf{x}, s_i, s_{i-1}, d_i)\}}{Z_{\mathbf{x},s_{i-1},d_i;\mathbf{w}_s}}$$

where,

$$Z_{\mathbf{x},s_{i-1},d_i;\mathbf{w}_s} = \sum_{s \in \mathcal{S}} \exp\{\mathbf{w}_s \cdot \phi_s(\mathbf{x}, s, s_{i-1}, d_i)\}$$

Similarly, given a feature map ϕ_d and a weight vector $\mathbf{w}_d$, ($p_{\mathbf{w}_d}(d_i)$ is short for $p_{\mathbf{w}_d}(d_i|s_{i-1}, d_{i-1}; \mathbf{x})$

$$p_{\mathbf{w}_d}(d_i) = \frac{\exp\{\mathbf{w}_d \cdot \phi_d(\mathbf{x}, d_i, s_{i-1}, d_{i-1})\}}{Z_{\mathbf{x},s_{i-1},d_{i-1};\mathbf{w}_d}}$$

where,

$$Z_{\mathbf{x},s_{i-1},d_{i-1};\mathbf{w}_d} = \sum_{d \in \mathcal{D}} \exp\{\mathbf{w}_d \cdot \phi_d(\mathbf{x}, d, s_{i-1}, d_{i-1})\}$$

For different characteristic properties between syntactic parsing and semantic parsing, different feature maps are taken into account. Table 2 lists the features used to predict semantic dependency relations, whereas table 3 lists the features used to predict the syntactic dependency relations. The features used for syntactic dependency relation classification are strongly based on previous works (McDonald et al., 2006; Nakagawa, 2007).

We just integrate syntactic dependency Relation classification and semantic dependency relation here. If one combines identification and classification of semantic roles as one multi-class classification, the tag set of the second layer can be substituted by the tag set of semantic roles plus a NULL ("not an argument") label.

3.4 Inference

The "argmax problem" in structured prediction is not tractable in the general case. However, the bilayer graphical model presented in form sections admits efficient search using dynamic programming solution. Searching for the highest probability of a graph depends on the factorization chosen. According to the form of the global score

$$p(\mathbf{d}, \mathbf{s}|\mathbf{x}) = \prod_{i=1}^{n} p(s_i|s_{i-1}, d_i; x) p(d_i|s_{i-1}, d_{i-1}; x)$$

, we define forward probabilities $\alpha_t(s, d)$ to be the probability of semantic relation being s and syntactic relation being d at time t given observation sequence up to time t. The recursive dynamic programming step is

$$\begin{aligned}\alpha_{t+1}(d, s) \quad = \quad & \arg \max_{d \in \mathcal{D}, s \in \mathcal{S}} \sum_{d' \in \mathcal{D}, s' \in \mathcal{S}} \alpha_t(d', s') \cdot \\ & p(s_i|s_{i-1}, d_i; \mathbf{x}) p(d_i|s_{i-1}, d_{i-1}; \mathbf{x})\end{aligned}$$

Finally, to compute the globally most probable assignment $(\hat{\mathbf{d}}, \hat{\mathbf{s}}) = \arg\max_{\mathbf{d},\mathbf{s}} p(\mathbf{d}, \mathbf{s}|\mathbf{x})$, a Viterbi recursion works well.

4 Results

We trained our system using positive examples extracted from all training data of CoNLL 2008 shared task. Table 4 shows the overall syntactic parsing results obtained on the WSJ test set (Section 23) and the Brown test set (Section ck/01-03). Table 5 shows the overall semantic parsing results obtained on the WSJ test set (Section 23) and the Brown test set (Section ck/01-03).

Test Set	UAS	LAS	Label Accuracy
WSJ	89.25%	86.37%	91.25%
Brown	86.12%	80.75%	87.14%

Table 4: Overall syntactic parsing results

	Task	Precision	Recall	$F_{\beta=1}$
WSJ	ID	73.76%	85.24%	79.08
	ID&CL	63.07%	72.88%	67.62
Brown	ID	70.77%	80.50%	75.32
	ID&CL	54.74%	62.26%	58.26

Table 5: Overall semantic parsing results

Test WSJ	Precision(%)	Recall(%)	$F_{\beta=1}$
SRL of Verbs			
All	73.53	73.28	73.41
Core-Arg	78.83	76.93	77.87
AM-*	62.51	64.83	63.65
SRL of Nouns			
All	62.06	45.49	52.50
Core-Arg	61.47	46.56	52.98
AM-*	66.19	39.93	49.81

Table 6: Semantic role labeling results on verbs and nouns. *Core-Arg* means numbered argument.

Table 6 shows the detailed semantic parsing results obtained on the WSJ test set (Section 23) of verbs and nouns respectively. The comparison suggests that SRL on NomBank is much harder than PropBank.

Acknowlegements

The work is supported by the National Natural Science Foundation of China under Grants No. 60503071, 863 the National High Technology Research and Development Program of China under Grants No.2006AA01Z144, and the Project of Toshiba (China) Co., Ltd. R&D Center.

References

Berger, Adam, Stephen Della Pietra, and Vincent Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computional Linguistics*, 22(1):39–71.

Ge, Ruifang and Raymond J. Mooney. 2005. A Statistical Semantic Parser that Integrates Syntax and Semantics. In *Proceedings of the Conference of Computational Natural Language Learning*.

Gildea, Daniel and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computional Linguistics*, 28(3):245–288.

Koomen, Peter, Vasina Punyakanok, Dan Roth, and Wen-tau Yih. 2005. Generalized Inference with Multiple Semantic Role Labeling Systems. In *Proceedings of Conference on Natural Language Learning*.

McCallum, Andrew, Dayne Freitag, and Fernando Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of International Conference on Machine Learning*.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

McDonald, Ryan, Kevin Lerman, and Fernando Pereira. 2006. Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In *Proceedings of Conference on Natural Language Learning*.

Nakawa, Tetsuji. 2007. Multilingual Dependency Parsing using Global Features. In *Proceedings of Conference on Natural Language Learning*.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 Shared Task on Dependency Parsing. 2007. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, 915–932,

Pradhan, Sameer, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James Martin, and Daniel Jurafsky. 2005. Support Vector Learning for Semantic Argument Classification. In *Proceedings of Conference on Association for Computational Linguistics*.

Punyakanok, Vasin , Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic Role Labeling via Integer Linear Programming Inference. In *Proceedings of the 20th International Conference on Computational Linguistics*.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez, and Nivre, Joakim. 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)*.

Xue, Nianwen and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling. In *Proceedings of Empirical Methods in Natural Language Processing*.

Yi, Szu-ting and Martha Palmer. 2005. The Integration of Syntactic Parsing and Semantic Role Labeling. In *Proceedings of the Conference of Computational Natural Language Learning*.