Using Word Sense Disambiguation for Semantic Role Labeling

Wanxiang Che and Ting Liu
Research Center for Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China 150001
Email: {car, tliu}@ir.hit.edu.cn

Abstract—**Word sense disambiguation (WSD) is the process of identifying the correct meaning, or sense of a word in a given context. Semantic role labeling (SRL) aims at identifying the relations between predicates in a sentence and their associated arguments. They are two fundamental tasks in natural language processing to find a sentence-level semantic representation. To date, they have mostly been modeled in isolation. However, this approach neglects logical constraints between them. In this work, we present some novel word sense features for SRL and find that they can improve the performance significantly. Then, we exploit pipeline strategies which verify the automatic all word sense disambiguation could help the SRL. We further propose a Markov logic model that jointly labels semantic roles and disambiguates all word senses. The experiments show that this joint approach leads to a higher performance for SRL than the pipeline approaches.**

I. INTRODUCTION

Semantic role labeling (SRL) and word sense disambiguation (WSD) are two fundamental tasks in natural language processing to find a sentence-level semantic representation. SRL aims at identifying the relations between predicates in a sentence and their associated arguments. WSD is the process of identifying the correct meaning, or sense of a word in a given context. For example, for the sentence in Figure 1, we can find out that the predicate token "hitting" at position 3 has sense "cause to move by striking" and the sense label is "hit.01". The argument headed by the token "cat" at position 1 with sense "feline mammal" (cat.01) is referring to the player (A0), and the argument headed by the token "ball" at position 5 with sense "round object that is hit in games" (ball.01) is referring to the game object (A1) being hit.
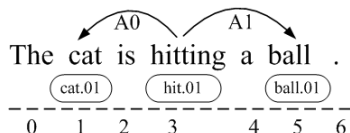

Fig. 1. A sample of WSD and SRL.

Word senses are important information for recognizing semantic roles. For example, if we know "cat" is an "agent" of the predicate "hit" in a sentence, we can guess that "dog" can also be an "agent" of "hit". Word sense has been successfully used in many natural language processing tasks, such as machine translation [1], [2]. CoNLL 2008 shared task [3] first introduced the predicate classification task, which can be regarded as the predicate sense disambiguation. [4] has shown that the predicate sense can improve the final SRL performance. However, there is few discussion about the concrete influence of all word senses, i.e., the words besides predicates.

On the other hand, SRL and WSD are always regarded as two independent tasks, i.e., the word sense information is rarely used in an SRL system and vice versa. Some researchers used predicate senses in SRL [5], [3]. However, the pipeline methods ignore possible dependencies between the word senses and semantic roles, and can result in the error propagation problem. The same problem also appears in other natural language processing tasks.

In order to make different natural language processing tasks be able to help each other, jointly modeling methods become popular recently, such as joint Chinese word segmentation and part-of-speech tagging [6], [7], [8], joint lemmatization and part-of-speech prediction [9], joint morphological segmentation and syntactic parsing [10], joint text and aspect ratings for sentiment summarization [11], joint parsing and named entity recognition [12], joint tokenization and translation [13], and joint parsing and translation [14]. For SRL, [15] proposed a method to maximize the joint probability of the semantic role of preposition phrases and the preposition sense.

In order to do better joint learning, a novel statistical relational learning framework, Markov logic [16] was introduced to join SRL and predicate senses [4]. Markov logic combines the first order logic and Markov networks, to develop a joint probability model over all related rules. Global constraints among semantic roles can be easily added into Markov logic. More importantly, the jointly modeling can be realized using Markov logic naturally. Besides predicates and prepositions, other word senses are also important information for recognizing semantic roles. Therefore, we use all word sense disambiguation for SRL.

The release of OntoNotes [17] provides us an opportunity to jointly model all word senses disambiguation and SRL. OntoNotes is a large corpus annotated with constituency

trees (based on Penn Treebank), predicate argument structures (based on Penn PropBank), all word senses, etc. It has been used in some natural language processing tasks, such as joint parsing and named entity recognition [12], and WSD [18].

In this paper, we first present an SRL system with additional golden word sense information in order to find optimal word sense features. Then, we propose some pipeline systems which exploit automatic all word sense disambiguation into SRL task. Then we present a Markov logic model which can easily express useful global constraints and jointly disambiguate all word senses and label semantic roles.

II. RELATED WORK

Joint models were often used in SRL community. [19] and [20] presented a re-ranking model and an integer linear programming model respectively to jointly learn a global optimal semantic roles assignment. Besides jointly learning semantic role assignment of different constituents for one task (SRL), their methods have been used to jointly learn for two tasks (SRL and syntactic parsing). However, it is easy for the re-ranking model to loss the optimal result, if it is not included in the top n results. In addition, the integer linear programming model can only use hard constraints. A lot of engineering work is also required in both models.

Recently, Markov logic [16] became a hot framework for joint model. It has been successfully used in temporal relations recognition [21], co-reference resolution [22], etc. It is very easy to do joint modeling using Markov logic. The only work is to define relevant formulas. [4] have joined SRL and predicate senses disambiguation with Markov logic.

The above idea, that the predicate senses and the SRL can help each other, may be inspired by [5] and [3]. They have shown that the verb sense features are helpful to label semantic roles.

Besides predicate senses, [15] proposed a joint model to maximize probability of the preposition senses and the semantic role of prepositional phrases.

III. SRL SYSTEM

Our baseline is a state-of-the-art SRL system based on dependency syntactic tree [23], which obtained the first place in the ConLL 2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. A maximum entropy [24] classifier is used to predict the probabilities of a word in the sentence to be each semantic role. A virtual role "NULL" (presenting none of roles is assigned) is added to the roles set, so it does not need semantic role identification stage anymore. For a predicate, two classifiers (one for noun predicates, and the other for verb predicates) predict probabilities of each word in a sentence to be each semantic role (including virtual role "NULL"). The features used in this stage are listed in Table I.

The probability of each word to be a semantic role for a predicate is given by the above stage. The results generated by selecting the roles with the largest probabilities, however, do not satisfy some constrains. An integer linear programming

(ILP) [25] method is used to get the global optimization, which is satisfied with two constrains:

C1: Each word should be labeled with one and only one label (including the virtual label "NULL").

C2: Some roles (A0∼A5) appear once for a predicate.

IV. WORD SENSE FOR SRL

From Table I, we can see that there are lots of lemma or POS related features. However, the lemma feature is very sparse and may result in data sparseness problem. As for the POS, it represents the syntactic information, but is not enough to distinguish different semantic roles. Therefore, we need a kind of new feature, which is general than the lemma and special than the POS.

The word sense just satisfies the requirement. Thus, we will add some new features related with word sense for SRL. Generally, the original features can be classified into three categories:

1) Argument related: FirstwordLemma, LastwordLemma, HeadwordLemma, and HeadwordPOS
2) Predicate related: PredicateLemma
3) Word and predicate related: POSPath, RelationPath, PathLenght, and Position

Correspondingly, we add three categories of word sense features by replacing Lemma or POS into Sense, i.e.,

1) Argument related sense: FirstwordSense, LastwordSense, and HeadwordSense
2) Predicate related sense: PredicateSense
3) Word and predicate related sense: SensePath

Three strategies are designed to adopt these senses:

1) Lemma+Sense: It is the original word sense representation in OntoNotes, such as "dog.n.1". In fact, This is a specialization of the lemma.
2) Hypernym(n): It is the hypernym of a word sense, e.g. the hypernym of "dog.n.1" is "canine.n.1". The n means the level of the hypernym. With the increasing of n, the sense becomes more and more general. In theory, however, this strategy may result in inconsistent sense, e.g. word "dog" and "canine" have different hypernyms. The same problem occurs with Basic Concepts method [26].

Feature	Description
FirstwordLemma	The lemma of the first word in a subtree
HeadwordLemma	The lemma of the head word in a subtree
HeadwordPOS	The POS of the head word in a subtree
LastwordLemma	The lemma of the last word in a subtree
POSPath	The POS path from a word to a predicate
PathLength	The length of a path
Position	The relative position of a word with a predicate
PredicateLemma	The lemma of a predicate
RelationPath	The dependency relation path from a word to a predicate

TABLE I
FEATURES THAT ARE USED IN SRL.

3) Root_Hyper(n): In order to extract more consistent sense, we use the hypernym of a word sense counting from the root of a sense tree, e.g. the root hypernym of "dog.n.1" is "entity.n.1". The n means the level of the root hypernym. With the increasing of n, the sense becomes more and more special. Thus, word "dog" and "canine" have the same Root_Hyper: "entity", "physical_entity", and "object" with $n = 1$, 2, and 3 respectively.

V. MARKOV LOGIC

Markov logic can be understood as a knowledge representation with a weight attached to a first-order logic formula. Let us describe Markov logic in the case of the SRL task. We can model this task by first introducing a set of logical predicates such as $role(p, a, r)$ and $lemma(i, l)$, which means that the argument at position a has the role r with respect to the predicate at position p and token at position i has lemma l respectively. Then we specify a set of weighted first order formulas that define a distribution over sets of ground atoms of these predicates (or so-called possible worlds).

Ideally, the distribution we define with these weighted formulas assigns high probability to possible worlds where SRL is correct and a low probability to worlds where this is not the case. For instance, for the sentence in Figure 1, a suitable set of weighted formulas would assign a high probability to the world:

$$lemma(1, cat), lemma(3, hit), lemma(5, ball)$$
$$role(3, 1, A0), role(3, 5, A1)$$

and low probabilities to other cases.

A Markov logic network (MLN) M is a set of weighted formulas, i.e., a set of pairs (ϕ, ω), where ϕ is a first order formula and ω is the real weight of the formula. M defines a probability distribution over possible worlds:

$$p(y) = \frac{1}{Z} \exp\left(\sum_{(\phi,\omega)\in M} \omega \sum_{c\in C^\phi} f_c^\phi(y) \right)$$

where each c is a binding of free variables in ϕ to constants. Each f_c^ϕ is a binary feature function that returns 1 if the possible world y includes the ground formula by replacing the free variables in ϕ with the constants in c is true, and 0 otherwise. C^ϕ is the set of all bindings for the variables in ϕ. Z is a normalization constant.

VI. MODEL

We divide our system into two stages: WSD and SRL. For comparison, we can process them with pipeline strategy, i.e., the WSD results are used in SRL. Of course, we can jointly process them with Markov logic easily.

We define two hidden predicates for the two stages respectively. For WSD, we define the predicate $sense(w, s)$ which indicates that the word at position w has the sense s. For SRL, the predicate $role(p, a, r)$ is defined as mentioned above.

In addition to the hidden predicates, we define observable predicates to represent the information available in the corpus. Table II presents these predicates.

Predicates	Description
$word(i, w)$	Token i has word w
$pos(i, t)$	Token i has part-of-speech t
$lemma(i, l)$	Token i has lemma l
$chdpos(i, t)$	The part-of-speech string of token i's all children is t
$chddep(i, d)$	The dependency relation string of token i's all children is t
$firstLemma(i, l)$	The leftmost lemma of a sub-tree rooted by token i is l
$lastLemma(i, l)$	The rightmost lemma of a sub-tree rooted by token i is l
$posFrame(i, fr)$	fr is a part-of-speech frame at token i
$dep(h, a, de)$	The dependency relation between an argument a and its head h is de
$isPredicate(p)$	Token p is a predicate
$posPath(p, a, pa)$	The part-of-speech path between a predicate p and an argument a is pa
$depPath(p, a, pa)$	The dependency relation path between a predicate p and an argument a is pa
$pathLen(p, a, le)$	The path length between a predicate p and an argument a is le
$position(p, a, po)$	The relative position between a predicate p and an argument a is po
$family(p, a, fa)$	The family relation between a predicate p and an argument a is fa
$wsdCand(i, t)$	Token i is a WSD candidate, here t is "v" or "n"
$uniqe(r)$	For a predicate, semantic role r can only appear once

TABLE II
OBSERVABLE PREDICATES.

A. Local Formula

A local formula means that its groundings relate any number of observed ground atoms to exactly one hidden ground atom. For example

$$lemma(p, +l_1) \land lemma(a, +l_2) \Rightarrow role(p, a, +r)$$

means that if the predicate lemma at position p is l_1 and the argument lemma at position a is l_2, then the semantic role between the predicate and the argument is r with some possibility.

The $+$ notation signifies that Markov logic generates a separate formula and a separate weight for each constant of the appropriate type, such as each possible pair of lemmas (l_1, l_2, r). This type of "template-based" formula generation can be performed automatically by a Markov logic engine, such as the *thebeast*[1] system.

The local formulas are based on features employed in the state-of-the-art systems. For WSD, we use the basic features mentioned by [18]. The SRL features are from Section III. The final features are listed in Table III.

What follows are some simple examples in order to explain how we implement each feature as a formula (or a set of

---

[1]http://code.google.com/p/thebeast/

Features	SRL	WSD
Lemma	•	•
POS	•	•
FirstwordLemma	•	
HeadwordLemma	•	
HeadwordPOS	•	
LastwordLemma	•	
POSPath	•	
PathLength	•	
Position	•	
PredicateLemma	•	
PredicatePOS	•	
RelationPath	•	
DepRelation	•	
POSUpPath	•	
POSFrame	•	
FamilyShip	•	
BagOfWords		•
Window3OrderedWords		•
Window3OrderedPOSs		•

TABLE III
LOCAL FEATURES.

formulas).

Consider the "Position" feature. We first introduce a predicate $position(p, a, po)$ that denotes the relative position between predicate p and argument a is po. Then we add a formula

$$position(p, a, +po) \Rightarrow role(p, a, +r)$$

for all possible combinations of $position$ and $role$ relations.

The "BagOfWords" feature means that the sense of a word w is determined by all of lemmas in a sentence. Then, we add the following formula set:

$$wsdCand(w, +t_w) \wedge lemma(w, +l_w) \wedge lemma(1, +l_1) \Rightarrow sense(w, +s)$$
$$\cdots$$
$$wsdCand(w, +t_w) \wedge lemma(w, +l_w) \wedge lemma(2, +l_i) \Rightarrow sense(w, +s)$$
$$\cdots$$
$$wsdCand(w, +t_w) \wedge lemma(w, +l_w) \wedge lemma(n, +l_n) \Rightarrow sense(w, +s)$$

where, the w is the position of current word and t_w is its part-of-speech tag, l_w is its lemma. l_i is the lemma of token i. There are n tokens in a sentence totally.

B. Global Formula

Global formulas relate more than one hidden ground atoms. We use this type of formula for two purposes:

1. To capture the global constraints among different semantic roles;

2. To reflect the joint relation between WSD and SRL.

Section III proposed an integer linear programming (ILP) model to get the global optimization for SRL, which satisfies some constraints. The same constraints can be described with Markov logic formulas as follows:

C1: Each word should be labeled with one and only one label.

$$role(p, a, r_1) \wedge r_1 \neq r_2 \Rightarrow \neg role(p, a, r_2)$$

The same unique constraint also happens on the WSD, i.e.,

$$sense(w, s_1) \wedge s_1 \neq s_2 \Rightarrow \neg sense(w, s_2)$$

C2: Some roles (A0∼A5) appear only once for a predicate.

$$role(p, a_1, r) \wedge uniqe(r) \wedge a_1 \neq a_2 \Rightarrow \neg role(p, a_2, r)$$

It is also easy to express the joint relation between WSD and SRL with Markov logic. What we need to do is just adding some global formulas. The relation between them can be shown in Figure 2. Inspired by CoNLL 2008 [3] and 2009 [5] shared tasks, where most of successful participant systems used predicate senses for SRL, we also model that the WSD implicates the SRL.

Here, we divide the all word sense disambiguation task into two subtasks: predicate sense disambiguation and argument sense disambiguation. The advantage of the division method approach lie in that makes us distinguish the contributions of predicate and argument WSD respectively.
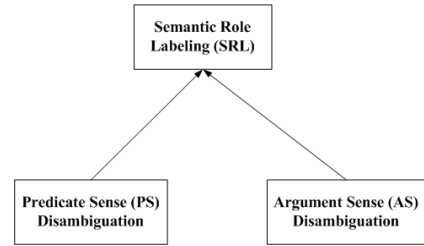

Fig. 2. Global model between WSD and SRL.

Finally, we use three global formulas to implement the two lines with direction in Figure 2. They are:

$$sense(p, +s) \Rightarrow role(p, a, +r)$$
$$sense(a, +s) \Rightarrow role(p, a, +r)$$

VII. EXPERIMENTS

A. Experimental Setting

In our experiments, we use the OntoNotes Release 3.0[2] corpus, the latest version of OntoNotes [17]. The OntoNotes project leaders describe it as "a large, multilingual richly-annotated corpus constructed at 90% internanotator agreement." The corpus has been annotated with multiple levels of annotation, including constituency trees, predicate argument structure, word senses, co-reference, and named entities. For this work, we focus on the constituency trees, word senses, and predicate argument structures. The corpus has English, Chinese, and Arabic portions, and we just use the English portion, which has been split into four sections: broadcast conversation (bc), broadcast news (bn), magazine (mz), and newswire (nw). There are several datasets in each section, such as cnn and voa.

We will do our experiments on all of the OntoNotes 3.0 English datasets. For each dataset, we aimed for roughly a 60% train / 20% development / 20% test split. See Table IV for the detailed statistics. Here, we use the human annotated

[2]http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?
catalogId=LDC2009T24

		Training		Developing		Testing	
bc	cctv	1,042	(0000-0003)	328	(0004-0004)	333	(0005-0005)
	cnn	2,927	(0000-0004)	963	(0005-0006)	880	(0007-0008)
	msnbc	2,472	(0000-0003)	1,209	(0004-0005)	1,315	(0006-0007)
	phoenix	590	(0000-0001)	240	(0002-0002)	322	(0003-0003)
bn	abc	594	(0001-0040)	146	(0041-0054)	126	(0057-0069)
	cnn	1,610	(0001-0234)	835	(0235-0329)	1,068	(0330-0437)
	mnb	309	(0001-0015)	111	(0016-0020)	114	(0021-0025)
	nbc	281	(0001-0023)	128	(0024-0031)	78	(0032-0039)
	pri	1,104	(0001-0068)	399	(0069-0090)	366	(0091-0112)
	voa	1,159	(0001-0159)	315	(0160-0212)	315	(0213-0265)
mz	sinorama	5,051	(1001-1048)	1,262	(1049-1063)	1,456	(1064-1078)
nw	wsj	8,138	(0020-1446)	2,549	(1447-1705)	3,133	(1730-2454)
	xinhua	2,285	(0001-0195)	724	(0196-0260)	670	(0261-0325)
All		27,562		9,209		10,176	

TABLE IV

TRAINING, DEVELOPING AND TESTING SET SIZES FOR THE DATASETS IN SENTENCES. THE FILE RANGES (IN PARENTHESIS) REFER TO THE NUMBERS WITHIN THE NAMES OF THE ORIGINAL ONTONOTES 3.0 FILES. HERE, WE REMOVE 4,873 SENTENCES WITHOUT SRL ANNOTATION.

part-of-speech and parse trees provided by OntoNotes. The lemma of each word is extracted using WordNet tool[3].

Because our SRL system is based on dependence syntactic trees, we convert the constituency trees into dependence trees with an Constituent-to-Dependency Conversion Tool[4].

Our maximum entropy classifier is implemented with Maximum Entropy Modeling Toolkit[5] with the L-BFGS parameter estimation algorithm with gaussian prior smoothing. The *thebeast* system is used in our experiment as Markov logic engine. It uses cutting planes inference technique [27] with integer linear programming. The weights are learned with MIRA [28] online learning algorithm.

The performance of SRL is calculated using the semantic evaluation metric of the CoNLL 2009 shared task scorer[6]. It measures the precision, recall, and F_1 score of the recovered semantic dependencies. The F_1 score is used as the final performance metric. A semantic dependency is created for each predicate and its arguments. The label of such dependency is the role of the argument. The same with the CoNLL 2009 shared task, we assume that the predicates have been identified correctly.

B. Results and Discussion

1) SRL with Golden Word Sense: The baseline SRL system which is without sense information and trained with all the training corpus as described in Section III. Its performance on the development data is $F_1 = 84.95\%$.

Table V shows the performance (F_1) comparison on the development data among different sense extracting strategies with different feature categories. The numbers are the parameter n used in Hypernym and Root_Hyper strategies.

From Table V, we can find that:

1. Both of the predicate sense feature and the sense path feature can improve the performance.

[3] http://wordnet.princeton.edu/

[4] http://nlp.cs.lth.se/software/treebank_converter/

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

[6] http://ufal.mff.cuni.cz/conll2009-st/eval09.pl

		Argument sense	Predicate sense	Sense path
Lemma+Sense		85.04%	85.81%	85.40%
Hypernym(n)	1	85.11%	85.96%	85.38%
	2	85.15%	86.10%	85.58%
	3	85.15%	85.74%	85.47%
Root_Hyper(n)	1	85.08%	85.79%	85.63%
	2	85.03%	85.90%	85.74%
	3	85.03%	85.78%	85.76%

TABLE V

THE PERFORMANCE COMPARISON ON THE DEVELOPMENT DATA AMONG DIFFERENT SENSE EXTRACTING STRATEGIES WITH DIFFERENT FEATURE CATEGORIES.

2. The argument sense is almost useless. The reason is that the original lemma and POS features have been able to describe the argument related information. This kind of sense features is just reduplicate.

3. For different sense feature categories (columns), the performance is not very seriously affected by different sense extracting strategies (rows). That is to say, once the sense of a word is disambiguated, the sense expressing form is not important for SRL.

2) Joint WSD and SRL: To our knowledge, this is the first WSD and SRL experiment on OntoNotes 3.0 corpus. In order to compare our joint model with previous work, we build several systems:

ML Baseline: In the baseline systems, we use Markov logic and only use the local formulas (Section VI-A) and the global formulas which only express the global constraints (Section VI-B).

Pipeline: In the pipeline system, we use additional word sense features outputted by WSD stages, i.e., we set $sense(w, s)$ as an observable predicate and add $sense(p, s) \Rightarrow role(p, a, r)$ and $sense(a, s) \Rightarrow role(p, a, r)$ formulas into SRL task.

Jointly: We use all global formulas mentioned in Section VI-B. With Markov logic, we can add global constraints and get the SRL results simultaneously.

ML Baseline		84.21
Pipeline	PS	84.35
	AS	84.20
	PS + AS	84.44
Jointly	PS $\Rightarrow$ SRL	84.55
	PS + AS $\Rightarrow$ SRL	84.59

TABLE VI

THE RESULTS OF DIFFERENT SYSTEMS. HERE, PS MEANS PREDICATE
SENSES AND AS MEANS ARGUMENT SENSES.

The performances of these systems are shown in Table VI. All of the parameters are fine tuned on the development set.

Here, we only consider the noun and verb WSD, which cover most of multi-sense words.

The first row shows the baseline performances. Here, we note that the 84.21 SRL F_1 we obtained are comparable to the state-of-the-art system based on maximum entropy classifier. Therefore, the experimental results show that the Markov logic can achieve considerable performances for the SRL on the latest OntoNotes 3.0 corpus.

For the using method of word senses, we first only exploit predicate senses (PS) as mentioned by [3] and [5]. Then, in order to examine the contribution of word senses except for predicates, we use argument senses (AS) in isolation. Finally, all word senses (PS + AS) were considered.

We can see that when the predicate senses (PS) are used to label semantic role, the performance can be improved from 84.21 to 84.35. The conclusion, that the predicate sense can improve SRL performance, is similar with CoNLL 2008 [3] and 2009 [5] shared tasks. However, the improvement is not significant (χ^2-test[7] with $\rho < 0.1$).

However, when we only use argument senses (AS), the SRL performances are almost unchanged. This result further validates that the argument sense self is useless.

Therefore, we use the last strategy (PS + AS), which combines the predicate sense and the argument sense together to predict semantic roles. The results show that the performance can be improved significantly (χ^2-test with $\rho < 0.05$) from 84.21 to 84.44. Accordingly, the experiment proves that automatic all word sense disambiguation can further improve the SRL performance.

With Markov logic, we can easily join different tasks with global formulas. As shown in Table VI, we use two joint strategies:

1. PS $\Rightarrow$ SRL: means that we jointly disambiguate predicate senses and label semantic roles. Compared with the pipeline PS system, the performance is improved from 84.35 to 84.55. Compared with the baseline's 84.21, the improvement is significant (χ^2-test with $\rho < 0.05$).

2. PS + AS $\Rightarrow$ SRL: means that we jointly disambiguate all word senses and label semantic roles. Compared with the pipeline method which uses the PS + AS strategy, the joint method can further improve the performance (from 84.34 to

84.59). Additionally, it can obtain the predicate and argument senses together.

VIII. CONCLUSION

This is the first effort to adopt the all word sense features into SRL. Experiments show that the argument related sense features are ineffective, but the predicate sense and the sense path features can improve the performance significantly.

We further presented a Markov logic model that jointly models all word sense disambiguation and SRL. We got the following conclusions:

1. The baseline systems with Markov logic is competitive to the state-of-the-art SRL systems on OntoNotes 3.0 corpus.

2. The predicate sense disambiguation is beneficial to SRL. However, the automatic argument sense disambiguation itself is harmful to the task. It must be combined with the predicate sense disambiguation.

3. It is easy to implement the joint model of all word sense disambiguation and SRL with Markov logic. More importantly, the joint model can further improve the performance of the SRL than pipeline systems.

REFERENCES

[1] Y. S. Chan, H. T. Ng, and D. Chiang, "Word sense disambiguation improves statistical machine translation," in *Proceedings of ACL-2007*, Prague, Czech Republic, June 2007, pp. 33–40. [Online]. Available: http://www.aclweb.org/anthology/P07-1005

[2] M. Carpuat and D. Wu, "Improving statistical machine translation using word sense disambiguation," in *Proceedings of EMNLP/CoNLL-2007*, Prague, Czech Republic, June 2007, pp. 61–72. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1007

[3] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre, "The conll 2008 shared task on joint parsing of syntactic and semantic dependencies," in *Proceedings of CoNLL 2008*, August 2008, pp. 159–177. [Online]. Available: http://www.aclweb.org/anthology/W08-2121

[4] I. Meza-Ruiz and S. Riedel, "Jointly identifying predicates, arguments and senses using markov logic," in *Proceedings of NAACL 2009*, June 2009, pp. 155–163. [Online]. Available: http://www.aclweb.org/anthology/N/N09/N09-1018

[5] J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang, "The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages," in *Proceedings of CoNLL 2009: Shared Task*, June 2009, pp. 1–18. [Online]. Available: http://www.aclweb.org/anthology/W09-1201

[6] C. Kruengkrai, K. Uchimoto, J. Kazama, Y. Wang, K. Torisawa, and H. Isahara, "An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging," in *Proceedings of ACL-IJCNLP 2009*, August 2009, pp. 513–521. [Online]. Available: http://www.aclweb.org/anthology/P/P09/P09-1058

[7] Y. Zhang and S. Clark, "Joint word segmentation and POS tagging using a single perceptron," in *Proceedings of ACL 2008*, June 2008, pp. 888–896. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1101

[7]http://graphpad.com/quickcalcs/chisquared1.cfm

[8] W. Jiang, L. Huang, Q. Liu, and Y. Lü, "A cascaded linear model for joint chinese word segmentation and part-of-speech tagging," in *Proceedings of ACL 2008*, June 2008, pp. 897–904. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1102

[9] K. Toutanova and C. Cherry, "A global model for joint lemmatization and part-of-speech prediction," in *Proceedings of ACL-IJCNLP 2009*, August 2009, pp. 486–494. [Online]. Available: http://www.aclweb.org/anthology/P/P09/P09-1055

[10] Y. Goldberg and R. Tsarfaty, "A single generative model for joint morphological segmentation and syntactic parsing," in *Proceedings of ACL 2008*, June 2008, pp. 371–379. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1043

[11] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization," in *Proceedings of ACL 2008*, June 2008, pp. 308–316. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1036

[12] J. R. Finkel and C. D. Manning, "Joint parsing and named entity recognition," in *Proceedings of NAACL 2009*, June 2009, pp. 326–334. [Online]. Available: http://www.aclweb.org/anthology/N/N09/N09-1037

[13] X. Xiao, Y. Liu, Y. Hwang, Q. Liu, and S. Lin, "Joint tokenization and translation," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, August 2010, pp. 1200–1208. [Online]. Available: http://www.aclweb.org/anthology/C10-1135

[14] Y. Liu and Q. Liu, "Joint parsing and translation," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, August 2010, pp. 707–715. [Online]. Available: http://www.aclweb.org/anthology/C10-1080

[15] D. Dahlmeier, H. T. Ng, and T. Schultz, "Joint learning of preposition senses and semantic roles of prepositional phrases," in *Proceedings of EMNLP 2009*, August 2009, pp. 450–458. [Online]. Available: http://www.aclweb.org/anthology/D/D09/D09-1047

[16] P. Domingos and D. Lowd, *Markov Logic: An Interface Layer for Artificial Intelligence*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

[17] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: The 90% solution," in *Proceedings of NAACL 2006*, June 2006, pp. 57–60. [Online]. Available: http://www.aclweb.org/anthology/N/N06/N06-2015

[18] Z. Zhong, H. T. Ng, and Y. S. Chan, "Word sense disambiguation using OntoNotes: An empirical study," in *Proceedings of EMNLP 2008*, October 2008, pp. 1002–1010. [Online]. Available: http://www.aclweb.org/anthology/D08-1105

[19] K. Toutanova, A. Haghighi, and C. D. Manning, "A global joint model for semantic role labeling," *Computational Linguistics*, vol. 34, no. 2, 2008.

[20] V. Punyakanok, D. Roth, and W. tau Yih, "The importance of syntactic parsing and inference in semantic role labeling," *Computational Linguistics*, vol. 34, no. 2, 2008. [Online]. Available: http://l2r.cs.uiuc.edu/ danr/Papers/PunyakanokRoYi07.pdf

[21] K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto, "Jointly identifying temporal relations with markov logic," in *Proceedings of ACL-IJCNLP 2009*, August 2009, pp. 405–413. [Online]. Available: http://www.aclweb.org/anthology/P/P09/P09-1046

[22] H. Poon and P. Domingos, "Joint unsupervised coreference resolution with Markov Logic," in *Proceedings of EMNLP 2008*, October 2008, pp. 650–659. [Online]. Available: http://www.aclweb.org/anthology/D08-1068

[23] W. Che, Z. Li, Y. Li, Y. Guo, B. Qin, and T. Liu, "Multilingual dependency-based syntactic and semantic parsing," in *Proceedings of CoNLL 2009: Shared Task*, June 2009, pp. 49–54. [Online]. Available: http://www.aclweb.org/anthology/W09-1207

[24] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, 1996.

[25] V. Punyakanok, D. Roth, W.-t. Yih, and D. Zimak, "Semantic role labeling via integer linear programming inference," in *Coling-2004*, 2004.

[26] R. Izquierdo, A. Suárez, and G. Rigau, "Exploring the automatic selection of basic level concepts," in *Proceedings of RANLP-2007*, 2007.

[27] S. Riedel, "Improving the accuracy and efficiency of map inference for markov logic," in *Proceedings of UAI 2008*. AUAI Press, 2008, pp. 468–475.

[28] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *Journal of Machine Learning Research*, vol. 3, pp. 951–991, 2003.