Multilingual Semantic Role Labelling with Markov Logic

Ivan Meza-Ruiz[*] **Sebastian Riedel**[†‡]
[*]School of Informatics, University of Edinburgh, UK
[†]Department of Computer Science, University of Tokyo, Japan
[‡]Database Center for Life Science, Research Organization of Information and System, Japan
[*]I.V.Meza-Ruiz@sms.ed.ac.uk [†] sebastian.riedel@gmail.com

Abstract

This paper presents our system for the CoNLL 2009 Shared Task on Syntactic and Semantic Dependencies in Multiple Languages (Hajič et al., 2009). In this work we focus only on the Semantic Role Labelling (SRL) task. We use Markov Logic to define a joint SRL model and achieve the third best average performance in the closed Track for SRLOnly systems and the sixth including for both SRLOnly and Joint systems.

1 Markov Logic

Markov Logic (ML, Richardson and Domingos, 2006) is a Statistical Relational Learning language based on First Order Logic and Markov Networks. It can be seen as a formalism that extends First Order Logic to allow formulae that can be violated with some penalty. From an alternative point of view, it is an expressive template language that uses First Order Logic formulae to instantiate Markov Networks of repetitive structure.

In the ML framework, we model the SRL task by first introducing a set of logical predicates[1] such as *word(Token,Ortho)* or *role(Token,Token,Role)*. In the case of *word/2* the predicate represents a word of a sentence, the type *Token* identifies the position of the word and the type *Ortho* its orthography. In the case of *role/3*, the predicate represents a semantic role. The first token identifies the position of the predicate, the second the syntactic head of the argument and finally the type Role signals the semantic role label. We will refer to predicates such as *word/2* as *observed* because they are known in advance. In contrast, *role/3* is *hidden* because we need to infer it at test time.

With the ML predicates we specify a set of weighted first order formulae that define a distribution over sets of ground atoms of these predicates (or so-called *possible worlds*). A set of weighted formulae is called a *Markov Logic Network* (MLN). Formally speaking, an MLN M is a set of pairs (ϕ, w) where ϕ is a first order formula and w a real weight. M assigns the probability

$$p(\mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{(\phi,w)\in M} w \sum_{\mathbf{c}\in C^{\phi}} f^{\phi}_{\mathbf{c}}(\mathbf{y})\right) \quad (1)$$

to the possible world $\mathbf{y}$. Here C^{ϕ} is the set of all possible bindings of the free variables in ϕ with the constants of our domain. $f^{\phi}_{\mathbf{c}}$ is a feature function that returns 1 if in the possible world $\mathbf{y}$ the *ground formula* we get by replacing the free variables in ϕ by the constants in $\mathbf{c}$ is true and 0 otherwise. Z is a normalisation constant. Note that this distribution corresponds to a Markov Network (the so-called *Ground Markov Network*) where nodes represent ground atoms and factors represent ground formulae.

In this work we use 1-best MIRA (Crammer and Singer, 2003) Online Learning in order to train the weights of an MLN. To find the SRL assignment with maximal *a posteriori* probability according to an MLN and observed sentence, we use Cutting Plane Inference (CPI, Riedel, 2008) with ILP base solver. This method is used during both test time and the MIRA online learning process.

[1]In the cases were is not obvious whether we refer to SRL or ML predicates we add the prefix SRL or ML, respectively.

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task, pages 85–90,
Boulder, Colorado, June 2009.

2 Model

In order to model the SRL task in the ML framework, we propose four hidden predicates. Consider the example of the previous section:

argument/1 indicates the phrase for which its head is a specific position is an SRL argument. In our example *argument(2)* signals that the phrase for which the word in position 2 is its head is an argument (i.e., *Ms. Haag*).

hasRole/2 relates a SRL predicate to a SRL argument. For example, *hasRole(3,2)* relates the predicate in position 3 (i.e., *play*) to the phrase which head is in position 2 (i.e., *Ms. Haag*).

role/3 identifies the role for a predicate-argument pair. For example, *role(3,2,ARG0)* denotes the role $ARG0$ for the SRL predicate in the position 2 and the SRL argument in position 3.

sense/2 denotes the sense of a predicate at a specific position. For example, *sense(3,02)* signals that the predicate in position 3 has the sense *02*.

We also define three sets of observable predicates. The first set represents information about each token as provided in the shared task corpora for the closed track: *word* for the word form (e.g. *word*(3,plays)); *plemma/2* for the lemma; *ppos/2* for the POS tag; *feat/3* for each feature-value pair; *dependency/3* for the head dependency and relation; *predicate/1* for tokens that are predicates according to the "FILLPRED" column. We will refer to these predicates as the *token* predicates.

The second set extends the information provided in the closed track corpus: *cpos/2* is a coarse POS tag (first letter of actual POS tag); *possibleArg/1* is true if the POS tag the token is a potential SRL argument POS tag (e.g., PUNC is not); *voice/2* denotes the voice for verbal tokens based on heuristics that use syntactic information, or based on features in the FEAT column of the data. We will refer to these predicates as the *extended* predicates.

Finally, the third set represents dependency information inspired by the features proposed by Xue and Palmer (2004). There are two types of predicates in this set: *paths* and *frames*. Paths capture the dependency path between two tokens, and frames the subcategorisation frame for a token or a pair of tokens. There are directed and undirected versions of paths, and labelled (with dependency relations) and unlabelled versions of paths and frames. Finally, we have a frame predicate with the distance from the predicate to its head. We will refer to the paths and most of the frames predicates as the *path* predicates, while we will consider the *frame* predicates for a unique token part *token* predicates.

The ML predicates here presented are used within the formulae of our MLN. We distinguish between two types of formula: local and global.

2.1 Local formulae

A formula is local if its groundings relate any number of observed ground atoms to exactly one hidden ground atom. For example, a grounding of the local formula

$$lemma(p, +l_1) \wedge lemma(a, +l_2) \Rightarrow hasRole(p, a)$$

connects a hidden *hasRole/2* ground atom to two observed *plemma/2* ground atoms. This formula can be interpreted as the feature for the predicate and argument lemmas in the argument identification stage of a pipeline SRL system. Note that the "+" prefix indicates that there is a different weight for each possible pair of lemmas (l_1, l_2).

We divide our local formulae into four sets, one for each hidden predicate. For instance, the set for *argument/1* only contains formulae in which the hidden predicate is *argument/1*.

The sets for *argument/1* and *sense/2* predicates have similar formulae since each predicate only involves one token at time: the SRL argument or the SRL predicate token. The formulae in these sets are defined using only *token* or *extended* observed predicates.

There are two differences between the *argument/1* and *sense/2* formulae. First, the *argument/1* formulae use the *possibleArg/1* predicate as precondition, while the sense formulae are conditioned on the *predicate/1* predicate. For instance, consider the *argument/1* formula based on word forms:

$$word(a, +w) \wedge possibleArg(a) \Rightarrow argument(a),$$

and the equivalent version for the *sense/2* predicate:

$$word(p, +w) \wedge predicate(p) \Rightarrow sense(p, +s).$$

This means we only apply the *argument/1* formulae if the token is a potential SRL argument, and the *sense/2* formulae if the token is a SRL predicate.

The second difference is the fact that for the *sense/2* formulae we have different weights for each possible sense (as indicated by the $+s$ term in the second formula above), while for the *argument/1* formulae this is not the case. This follows naturally from the fact that *argument/1* do not explicitly consider senses.

Table 1 presents templates for the local formuale of *argument/1* and *sense/2*. Templates allow us to compactly describe the FOL clauses of a ML. The template column shows the body of a clause. The last two columns of the table indicate if there is a clause with the given body and $argument(i)$ (I) or $sense(i, +s)$ (S) head, respectively. For example, consider the first row: since the last two columns of the row are marked, this template expands into two formulae: $word(i, +w) \Rightarrow argument(i)$ and $word(i, +w) \Rightarrow sense(i, +s)$. Including the preconditions for each hidden predicate we obtain the following formulae:

$$possibleArg(i) \wedge word(i, +w) \Rightarrow argument(i)$$

and

$$predicate(i) \wedge word(i, +w) \Rightarrow sense(i, +s).$$

In the case of the template marked with a "*" sign, the parameters $\mathbf{P}$ and $\mathbf{I}$, where $\mathbf{P} \in \{ppos, plemma\}$ and $\mathbf{I} \in \{-2, -1, 0, 1, 2\}$, have to be replaced by any combination of possible values. Since we generate $argument$ and $sense$ formulae for this template, the row corresponds to 20 formulae in total.

Table 2 shows the local formuale for *hasRole/2* and *role/3* predicates, for these formulae we use *token*, *extended* and *path* predicates. In this case, these templates have as precondition the formula $predicate(p) \wedge possibleArg(a)$. This ensures that the formulae are only applied for SRL predicates and potential SRL arguments. In the table we include the values to replace the template parameters with. Some of these formulae capture a notion of distance between SRL predicate and SRL argument and are implicitely conjoined with a $distance(p, a, +d)$ atom. If a formulae exists both with and without $distance$ atom, we write *Both* in the "Dist" column; if it only exists with the $distance$ atom, we write *Only*, otherwise *No*.

Note that Tables 1 and 2 do not mention the feature information provided in the corpora because this information was not available for every language. We therefore group the formulae which consider the *feature/3* predicate into another a set we call *feature* formulae. This is the summary of these formulae:

$feat(p, +f, +v) \Rightarrow sense(p, +s)$
$feat(p, +f, +v) \Rightarrow argument(a)$
$feat(p, +f, +v1) \wedge feat(p, f, +v2) \Rightarrow hasRole(p, a)$
$feat(p, +f, +v1) \wedge feat(p, f, +v2) \Rightarrow role(p, a, +r)$

Additionally, we define a set of language specific formulae. They are aimed to capture the relations between argument and its siblings for the *hasRole/2* and *role/3* predicates. In practice in turned out that these formulae were only beneficial for the Japanese language. This is a summary of such formulae which we called *argument siblings*:

$dep(a, h, \_) \wedge dep(h, c, \_) \wedge ppos(a, +p1) \wedge ppos(c, +p2) \Rightarrow hasRole(p, a)$
$dep(a, h, \_) \wedge dep(h, c, \_) \wedge ppos(a, +p1) \wedge ppos(c, +p2) \Rightarrow role(p, a, +r)$
$dep(a, h, \_) \wedge dep(h, c, \_) \wedge plemma(a, +p1) \wedge ppos(c, +p2) \Rightarrow hasRole(p, a)$
$dep(a, h, \_) \wedge dep(h, c, \_) \wedge plemma(a, +p1) \wedge ppos(c, +p2) \Rightarrow role(p, a, +r)$

With these sets of formulae we can build specific MLNs for each language in the shared task. We group the formulae into the modules: *argument/1*,

Template	I	S
$word(i, +w)$	X	X
$\mathbf{P}(i + \mathbf{I}, +v)$*	X	X
$cpos(i + 1, +c_1) \wedge cpos(i - 1, +c_2)$	X	X
$cpos(i + 1, +c_1) \wedge cpos(i - 1, +c_2) \wedge cpos(i + 2, +c_3) \wedge cpos(i - 2, +c_4)$	X	X
$dep(i, \_, +d)$	X	X
$dep(\_, i, +d)$	X	X
$ppos(i, +o) \wedge dep(i, j, +d)$	X	X
$ppos(i, +o_1) \wedge ppos(j, +o_2) \wedge dep(i, j, +d)$	X	X
$ppos(j, +o_1) \wedge ppos(k, +o_2) \wedge dep(j, k, \_) \wedge dep(k, i, +d)$	X	X
$plemma(i, +l) \wedge dep(j, i, +d)$	X	X
$frame(i, +f)$	X	X
(Empty Body)		X

Table 1: Templates of the local formulae for *argument/1* and *sense/2*. I: head of clause is $argument(i)$, S: head of clause is $sense(i, +s)$

Template	Parameters	Dist.	H	R
$\mathbf{P}(p, +v)$	$\mathbf{P} \in S_1$	Both	X	X
$plemma(p, +l) \wedge ppos(a, +o)$		No	X	
$ppos(p, +o) \wedge plemma(a, +l)$		No	X	
$plemma(p, +l_1) \wedge plemma(a, +l_2)$		Only	X	X
$ppos(p, +o_1) \wedge ppos(a, +o_2)$		Only	X	
$ppos(p, +o_1) \wedge ppos(a + \mathbf{I}, +o_2)$	$\mathbf{I} \in \{-1, 0, 1\}$	Only	X	
$plemma(p, +l)$		Only		X
$voice(p, +e) \wedge lemma(a, +l)$		Only		X
$cpos(p, +c_1) \wedge cpos(p + \mathbf{I}, +c_2) \wedge cpos(a, +c_3) \wedge cpos(a + \mathbf{J}, c_4)$	$\mathbf{I}, \mathbf{J} \in \{-1, 1\}^2$	No	X	X
$ppos(p, +v_1) \wedge ppos(a, \text{IN}) \wedge dep(a, m, \_) \wedge \mathbf{P}(m, +v_2)$	$\mathbf{P} \in S_1$	No	X	X
$plemma(p, +v_1) \wedge ppos(a, \text{IN}) \wedge dep(a, m, \_) \wedge ppos(m, +v_2)$		No	X	X
$\mathbf{P}(p, a, +v)$	$\mathbf{P} \in S_2$	No	X	X
$\mathbf{P}(p, a, +v) \wedge plemma(p, +l)$	$\mathbf{P} \in S_3$	No	X	X
$\mathbf{P}(p, a, +v) \wedge plemma(p, +l_1) \wedge plemma(a, +l_2)$	$\mathbf{P} \in S_4$	No	X	X
$pathFrame(p, a, +t) \wedge plemma(p, +l) \wedge voice(p, +e)$		No	X	X
$pathFrameDist(p, a, +t)$		Only	X	X
$pathFrameDist(p, a, +t) \wedge voice(p, +e)$		Only	X	X
$pathFrameDist(p, a, +t) \wedge plemma(p, +l)$		Only	X	X
$\mathbf{P}(p, a, +v) \wedge plemma(a, +l)$	$\mathbf{P} \in S_5$	Only	X	X
$\mathbf{P}(p, a, +v) \wedge ppos(p, +o)$	$\mathbf{P} \in S_5$	Only	X	X
$pathFrameDist(p, a, +t) \wedge ppos(p, +o_1) \wedge ppos(a, +o_2)$		Only	X	X
$path(p, a, +t) \wedge plemma(p, +l) \wedge cpos(a, +c)$		Only	X	X
$dep(\_, a, +d)$		Only	X	X
$dep(\_, a, +) \wedge voice(p, +e)$		Only	X	X
$dep(\_, a, +d_1) \wedge dep(\_, p, +d_2)$		Only	X	X
$(EmptyBody)$		No	X	X

Table 2: Templates of the local formulae for *hasRole/2* and *role/3*. H: head of clause is $hasRole(p, a)$, R: head of clause is $role(p, a, +r)$ and $S_1 = \{ppos, plemma\}, S_2 = \{frame, unlabelFrame, path\}, S_3 = \{frame, pathFrame\}, S_4 = \{frame, pathFrame, path\}, S_5 = \{pathFrameDist, path\}$

hasRole/2, *role/3*, *sense/3*, *feature* and *argument siblings*. Table 3 shows the different configurations of such modules that we used for the individual languages. We omit to mention the *argument/1*, *hasRole/2* and *role/3* modules because they are present for all languages.

A more detailed description of the formulae can be found in our MLN model files.[2] They can be used both as a reference and as input to our Markov Logic Engine,[3] and thus allow the reader to easily reproduce our results.

Set	Feature	*sense/2*	Argument siblings
Catalan	Yes	Yes	No
Chinese	No	Yes	No
Czech	Yes	No	No
English	No	Yes	No
German	Yes	Yes	No
Japanese	Yes	No	Yes
Spanish	Yes	Yes	No

Table 3: Different configuration of the modules for the formulae of the languages.

2.2 Global formulae

Global formulae relate several hidden ground atoms. We use them for two purposes: to ensure consis-

[2] `http://thebeast.googlecode.com/svn/mlns/conll09`

[3] `http://thebeast.googlecode.com`

tency between the decisions of all SRL stages and to capture some of our intuition about the task. We will refer to formulae that serve the first purpose as *structural constraints*. For example, a structural constraint is given by the (deterministic) formula

$$role(p, a, r) \Rightarrow hasRole(p, a)$$

which ensures that, whenever the argument a is given a label r with respect to the predicate p, this argument must be an argument of a as denoted by *hasRole(p,a)*.

The global formulae that capture our intuition about the task itself can be further divided into two classes. The first one uses deterministic or *hard* constraints such as

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

which forbids cases where distinct arguments of a predicate have the same role unless the role describes a modifier.

The second class of global formulae is *soft* or non-deterministic. For instance, the formula

$$lemma(p, +l) \wedge ppos(a, +p)$$
$$\wedge hasRole(p, a) \Rightarrow sense(p, +f)$$

is a soft global formula. It captures the observation that the sense of a verb or noun depends on the type of its arguments. Here the type of an argument token is represented by its POS tag.

Table 4 presents the global formulae used in this model.

Structural constraints
$hasRole(p, a) \Rightarrow argument(a)$ $role(p, a, r) \Rightarrow hasRole(p, a)$
$argument(a) \Rightarrow \exists p.hasRole(p, a)$ $hasRole(p, a) \Rightarrow \exists r.role(p, a, r)$
Hard constraints
$role(p, a, r_1) \wedge r_1 \neq r_2 \Rightarrow \neg role(p, a, r_2)$ $sense(p, s_1) \wedge s_1 \neq s_2 \Rightarrow \neg sense(p, r_2)$ $role\,(p, a_1, r) \wedge \neg mod\,(r) \wedge a_1 \neq a_2 \Rightarrow \neg role\,(p, a_2, r)$
Soft constraints
$role\,(p, a_1, r) \wedge \neg mod\,(r) \wedge a_1 \neq a_2 \Rightarrow \neg role\,(p, a_2, r)$ $plemma(p, +l) \wedge ppos(a, +p) \wedge hasRole(p, a) \Rightarrow sense(p, +f)$ $plemma(p, +l) \wedge role(p, a, +r) \Rightarrow sense(p, +f)$

Table 4: Global formulae for ML model

Language	Devel	Test	Train time	Test time
Average	77.25%	77.46%	11h 29m	23m
Catalan	78.10%	78.00%	6h 11m	14m
Chinese	77.97%	77.73%	36h 30m	34m
Czech	75.98%	75.75%	14h 21m	1h 7m
English	82.28%	83.34%	12h 26m	16m
German	72.05%	73.52%	2h 28m	7m
Japanese	76.34%	76.00%	2h 17m	4m
Spanish	78.03%	77.91%	6h 9m	16m

Table 5: F-scores for in-domain in corpora for each language.

3 Results

For our experiments we use the corpora provided in the SRLonly track of the shared task. Our MLN is tested on the following languages: Catalan and Spanish (Taulé et al., 2008) , Chinese (Palmer and Xue, 2009), Czech (Hajič et al., 2006),[4] English (Surdeanu et al., 2008), German (Burchardt et al., 2006), Japanese (Kawahara et al., 2002).

Table 5 presents the F1-scores and training/test times for the development and in-domain corpora. Clearly, our model does better for English. This is in part because the original model was developed for English.

To put these results into context: our SRL system is the third best in the *SRLOnly* track of the Shared Task, and it is the sixth best on both *Joint* and *SRLOnly* tracks. For five of the languages the difference to the F1 scores of the best system is 3%. However, for German it is 6.19% and for Czech 10.76%. One possible explanation for the poor performance on Czech data will be given below. Note that in comparison our system does slightly better in terms of precision than in terms of recall (we have the fifth best average precision and the eighth average recall).

Table 6 presents the F1 scores of our system for the out of domain test corpora. We observe a similar tendency: our system is the sixth best for both *Joint* and *SRLOnly* tracks. We also observe similar large differences between our scores and the best scores for German and Czech (i.e., $> 7.5\%$), while for English the difference is relatively small (i.e., $< 3\%$).

[4]For training we use only sentences shorter than 40 words in this corpus.

Language	Czech	English	German
F-score	77.34%	71.86%	62.37%

Table 6: F-scores for out-domain in corpora for each language.

Finally, we evaluated the effect of the *argument siblings* set of formulae introduced for the Japanese MLN. Without this set the F-score is 69.52% for the Japanese test set. Hence *argument siblings* formulae improve performance by more than 6%.

We found that the MLN for Czech was the one with the largest difference in performance when compared to the best system. By inspecting our results for the development set, we found that for Czech many of the errors were of a rather technical nature. Our system would usually extract frame IDs (such as "play.02") by concatenating the lemma of the token and outcome of the *sense/2* prediction (for the "02" part). However, in the case of Czech some frame IDs are not based on the lemma of the token, but on an abstract ID in a vocabulary (e.g., "v-w1757f1"). In these cases our heuristic failed, leading to poor results for frame ID extraction.

4 Conclusion

We presented a Markov Logic Network that performs joint multi-lingual Semantic Role Labelling. This network achieves the third best semantic F-scores in the closed track among the SRLOnly systems of the CoNLL-09 Shared Task, and sixth best semantic scores among SRLOnly and Joint systems for the closed task.

We observed that the inclusion of features which take into account information about the siblings of the argument were beneficial for SRL performance on the Japanese dataset. We also noticed that our poor performance with Czech are caused by our frame ID heuristic. Further work has to be done in order to overcome this problem.

References

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of LREC-2006*, Genoa, Italy, 2006.

Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003. ISSN 1533-7928.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, and Zdeněk Žabokrtský. Prague dependency treebank 2.0, 2006.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL-2009)*, Boulder, Colorado, USA, 2009.

Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the LREC-2002*, pages 2008–2013, Las Palmas, Canary Islands, 2002.

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

Matt Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

Sebastian Riedel. Improving the accuracy and efficiency of map inference for markov logic. In *UAI '08: Proceedings of the Annual Conference on Uncertainty in AI*, 2008.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL-2008*, 2008.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of LREC-2008*, Marrakesh, Morroco, 2008.

Nianwen Xue and Martha Palmer. Calibrating features for semantic role labeling. In *EMNLP '04: Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing*, 2004.