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 predicates1 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 $(\phi, w)$ where $\phi$ is a first order formula and $w$ a real weight. $M$ assigns the probability

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

(1)

to the possible world $y$. Here $C^\phi$ is the set of all possible bindings of the free variables in $\phi$ with the constants of our domain. $f^\phi_c$ is a feature function that returns 1 if in the possible world $y$ the ground formula we get by replacing the free variables in $\phi$ by the constants in $c$ 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.
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)); **lemma** for the lemma; **pos** for the POS tag; **feature** for each feature-value pair; **dependency** for the head dependency and relation; **predicate** for tokens that are predicates according to the “FILL-PRED” column. We will refer to these predicates as the **token** predicates.

The second set extends the information provided in the closed track corpus: **cpos** is a coarse POS tag (first letter of actual POS tag); **possibleArg** is true if the POS tag the token is a potential SRL argument POS tag (e.g., PUNC is not); **voice** 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

\[
\text{lemma}(p, +l_1) \land \text{lemma}(a, +l_2) \Rightarrow \text{hasRole}(p, a)
\]

connects a hidden hasRole/2 ground atom to two observed lemma/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:

\[
\text{word}(a, +w) \land \text{possibleArg}(a) \Rightarrow \text{argument}(a),
\]

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

\[
\text{word}(p, +w) \land \text{predicate}(p) \Rightarrow \text{sense}(p, +s).
\]

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 formulae 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: \(\text{word}(i, +w) \Rightarrow \text{argument}(i)\) and \(\text{word}(i, +w) \Rightarrow \text{sense}(i, +s)\). Including the preconditions for each hidden predicate we obtain the following formulae:

\[
\text{possible} \text{Arg}(i) \land \text{word}(i, +w) \Rightarrow \text{argument}(i)
\]

and

\[
\text{predicate}(i) \land \text{word}(i, +w) \Rightarrow \text{sense}(i, +s).
\]

In the case of the template marked with a “\(*\)” sign, the parameters \(\text{P}\) and \(\text{I}\), where \(\text{P} \in \{\text{ppos, plemma}\}\) and \(\text{I} \in \{-2, -1, 0, 1, 2\}\), 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 formula for hasRole/2 and role/3 predicates, for these formulae we use token, extended and path predicates. In this case, these templates have as preconditions the formula \(\text{predicate}(p) \land \text{possible}\text{Arg}(a)\). 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 implicitly conjoined with a \(\text{distance}(p, a, +d)\) atom. If a formula exists both with and without distance atom, we write \text{Both} in the “Dist” column; if it only exists with the distance atom, we write \text{Only}, otherwise \text{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 set we call feature formulae. This is the summary of these formulae:

\[
\text{feat}(p, +f, +v) \Rightarrow \text{sense}(p, +s)
\]

\[
\text{feat}(p, +f, +v) \Rightarrow \text{argument}(a)
\]

\[
\text{feat}(p, +f, +v1) \land \text{feat}(p, f, +v2) \Rightarrow \text{hasRole}(p, a)
\]

\[
\text{feat}(p, +f, +v1) \land \text{feat}(p, f, +v2) \Rightarrow \text{role}(p, a, +r)
\]

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:

\[
\text{dep}(a, h, _) \land \text{dep}(h, c, _) \land \text{ppos}(a, +p1) \land \\
\text{ppos}(c, +p2) \Rightarrow \text{hasRole}(p, a)
\]

\[
\text{dep}(a, h, _) \land \text{dep}(h, c, _) \land \text{ppos}(a, +p1) \land \\
\text{ppos}(c, +p2) \Rightarrow \text{role}(p, a, +r)
\]

\[
\text{dep}(a, h, _) \land \text{dep}(h, c, _) \land \text{ppos}(c, +p2) \Rightarrow \text{hasRole}(p, a)
\]

\[
\text{dep}(a, h, _) \land \text{dep}(h, c, _) \land \text{ppos}(c, +p2) \Rightarrow \text{role}(p, a, +r)
\]

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,
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,+) and $S_1 = \{ppos,plemma\}, S_2 = \{frame,unlabelFrame, path\}, S_3 = \{frame, pathFrame\}, S_4 = \{frame, pathFrame, path\}, S_5 = \{pathFrameDist, path\}$

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

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

<table>
<thead>
<tr>
<th>Set</th>
<th>Feature</th>
<th>sense/2</th>
<th>Argument siblings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalan</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Chinese</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Czech</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>English</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>German</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Japanese</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Spanish</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
tendency 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

\[ \text{role}(p, a, r) \Rightarrow \text{hasRole}(p, a) \]

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

\[ \text{role}(p, a, r_1) \land r_1 \neq r_2 \Rightarrow \neg \text{role}(p, a, r_2) \]

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 nondeterministic. For instance, the formula

\[ \text{lemma}(p, +l) \land \text{ppos}(a, +p) \land \text{hasRole}(p, a) \Rightarrow \text{sense}(p, +f) \]

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.

### 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 (Hajčí et al., 2006),\(^4\) English (Surdeanu et al., 2008), German (Burchardt et al., 2006), Japanese (Kawahara et al., 2002).

Table 5 presents the F-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.

Table 4: Global formulae for ML model

<table>
<thead>
<tr>
<th>Language</th>
<th>Devel</th>
<th>Test</th>
<th>Train time</th>
<th>Test time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>77.25%</td>
<td>77.40%</td>
<td>11h 29m</td>
<td>23m</td>
</tr>
<tr>
<td>Catalan</td>
<td>78.10%</td>
<td>78.00%</td>
<td>6h 11m</td>
<td>14m</td>
</tr>
<tr>
<td>Chinese</td>
<td>77.97%</td>
<td>77.73%</td>
<td>36h 30m</td>
<td>34m</td>
</tr>
<tr>
<td>Czech</td>
<td>75.98%</td>
<td>75.75%</td>
<td>14h 21m</td>
<td>1h 7m</td>
</tr>
<tr>
<td>English</td>
<td>82.28%</td>
<td>83.34%</td>
<td>12h 26m</td>
<td>16m</td>
</tr>
<tr>
<td>German</td>
<td>72.05%</td>
<td>73.52%</td>
<td>2h 28m</td>
<td>7m</td>
</tr>
<tr>
<td>Japanese</td>
<td>76.34%</td>
<td>76.00%</td>
<td>2h 17m</td>
<td>4m</td>
</tr>
<tr>
<td>Spanish</td>
<td>78.03%</td>
<td>77.91%</td>
<td>6h 9m</td>
<td>16m</td>
</tr>
</tbody>
</table>

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

\(^4\)For training we use only sentences shorter than 40 words in this corpus.
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


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.


