Analysis of Joint Inference Strategies for the Semantic Role Labeling of Spanish and Catalan

Mihai Surdeanu[1], Roser Morante[2], Lluís Màrquez[3]

[1]Barcelona Media Innovation Center
[2]Tilburg University
[3]Technical University of Catalonia

Abstract. This paper analyzes two joint inference approaches for semantic role labeling: re-ranking of candidate semantic frames generated by one local model and combination of two distinct models at argument-level using meta learning. We perform an empirical analysis on two recently released corpora of annotated semantic roles in Spanish and Catalan. This work yields several novel conclusions: (a) the proposed joint inference strategies yield good results even under adverse conditions: small training corpora, only two individual models available for combination, minimal output available from the individual models; (b) stacking of the two joint inference approaches is successful, which indicates that the two inference models provide complementary benefits. Our results are currently the best for the identification of semantic role for Spanish and Catalan.

1 Introduction

Semantic Role Labeling (SRL) is the task of analyzing clause predicates in open text by identifying arguments and tagging them with semantic labels indicating the role they play with respect to the verb, as in:

[Mr. Smith]$_{Agent}$ *sent* [the report]$_{Object}$ to [me]$_{Recipient}$ [this morning]$_{Temporal}$

Such sentence–level semantic analysis allows to determine "who" did "what" to "whom", "when" and "where", and, thus, characterize the participants and properties of the *events* established by the predicates. This semantic analysis in the form of event structures is very interesting for a broad spectrum of NLP applications.

The work proposed in this paper fits in the framework of supervised learning with joint inference for SRL. We introduce a stacking architecture that exploits several levels of global learning: in the first level we deploy two base SRL models that exploit only information local to each individual candidate argument; in the second level we perform re-ranking of the candidate frames generated by the base models; and lastly, we combine the outputs of the two individual models (after re-ranking) using meta-learning and sentence-level information.

The combination/joint inference models we introduce are not novel in themselves: all state-of-the-art SRL systems (see, e.g., [1–4]) include some kind of combination to increase robustness and to gain coverage and independence from parse errors. One may combine: 1) the output of several independent SRL basic systems [2, 5], or 2) several outputs from the same SRL system obtained by changing input annotations or other internal parameters [4, 3]. The combination can be as simple as selecting the best among

the set of complete candidate solutions, but usually consists of combining fragments of alternative solutions to construct the final output. Finally, the combination component may or may not involve machine learning. So far, most of the SRL work has been performed on English, but recently, there have been remarkable efforts in other languages. The work by [6] studies semantic role labeling for Chinese, using the Chinese PropBank and NomBank corpora. Also, SemEval-2007 featured the first evaluation exercise of SRL systems for languages other than English, namely for Spanish and Catalan [7].

Nevertheless, our approach has several novel issues. First, we show that the global inference strategies analyzed perform well even under unfavorable training conditions: the training corpora are small and the global models have access to limited information (only two models available for combination, no output probabilities provided). We show that a crucial condition for the success of the joint inference models is the design of a feature set with low sparsity. We propose such feature sets for both the re-ranking and combination models and also show that some features previously proposed for English SRL –i.e., syntactic and lexical features extracted from the local models– are harmful in our setup. A second novelty of this work is the stacking architecture proposed: to our knowledge, this is the first work that provides empirical proof that stacking of several joint inference approaches is a successful strategy for SRL.

The paper is organized as follows. Section 2 overviews the proposed strategy. Section 3 describes the local SRL models used. Section 4 introduces the two joint inference approaches analyzed in this paper. We evaluate the whole stacking strategy in Section 5. Section 6 concludes the paper.

2 Approach Overview

The strategy introduced in this paper stacks two joint inference components on top of two individual SRL models. The intuition behind our approach is that we compensate for the small training corpus by taking advantage of information typically not available to the independent argument classifiers, i.e., global information available at frame and sentence level and redundancy between individual models. We detail the proposed approach in Figure 1.

The first layer in our system consists of two SRL models, Model 1 and Model 2. We call these models *local* because they classify each argument independently of the other arguments in the same frame or sentence. Each local model is followed by a re-ranking component, which re-scores candidate frames –i.e., complete sequences of arguments for one predicate– according to their properties. The re-ranking model performs *joint* or *global* inference because its re-ranking scores depend on joint properties of the set of arguments in one frame. We currently have implemented the re-ranking component for one local model (Model 1). Nevertheless, this is sufficient to prove one of our main claims, i.e., that stacking global models is a successful strategy even when only a small amount of training data is available. The post-processing steps implement various corrections of the local model outputs, e.g., here we implement a series of patterns to capture locative and temporal modifier arguments that are missed by the local classifiers. In our architecture, the re-ranking component is placed before post-processing because the re-ranking classifier requires the output probabilities generated by the local models and these are no longer consistent after the post-processing corrections. The proposed

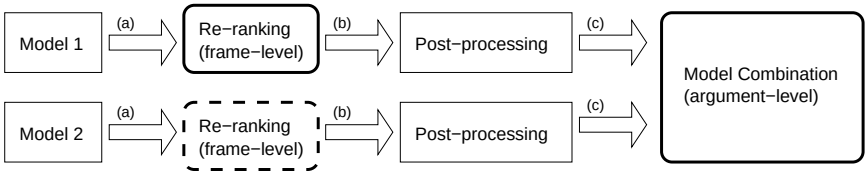

Fig. 1. Overview of the stacking architecture. The rounded boxes indicate joint inference components. The interrupted lines indicate components currently not implemented.

system pipeline concludes with another global model, which combines the outputs of the two branches. The combination model merges all the arguments generated for one sentence into one pool and re-scores them exploiting the redundancy of the two models and global information from the corresponding sentence.

3 Local Models

3.1 Model 1

Model 1 is an adaption of a SRL system we developed previously for English (third local model in [2]). This SRL approach maps each frame argument to one syntactic constituent and trains one-vs-all AdaBoost [8] classifiers to jointly identify and classify constituents in the full syntactic tree of the sentence as arguments. Similarly to other state-of-the-art SRL systems, our model extracts features from: (a) the argument constituent, (b) the target predicate, and (c) the relation between the predicate and argument syntactic constituents [9–12, 3]. Features range from lexical –e.g., head words of the argument and predicate constituents– to syntactic –e.g., constituent labels and syntactic path between the predicate and argument constituents.

The model was adapted to the Spanish and Catalan corpora by removing the features that were specific either to English or PropBank and adding several new features:

- We removed the *governing category* feature [9] because it does not apply to the Spanish and Catalan corpora: in PropBank, agents are typically dominated by a `S` (sentence) phrase, whereas patients are attached to `VP` (verb) phrase. In our corpora both arguments are dominated by the `S` phrase that includes the predicate.
- We removed *temporal cue words* features [13] because they were based on an English-specific dictionary.
- We removed all features based on named-entity information [10] because we did not have a named-entity recognizer for the target languages.
- We implemented head phrase selection heuristics for Spanish and Catalan.
- We added *syntactic function* features. The syntactic functions available in the data often point to specific argument labels (e.g., the function `SUJ` usually indicates an `Arg0` argument).
- Because the set of part-of-speech (POS) tags and syntactic labels in Spanish and Catalan is much richer than the English Treebank set, we added *back-off* features, where the syntactic labels and POS tags are reduced to a simpler, Treebank-like set.

All the other features are similar to the English SRL system described in [2]. In addition to feature changes we implemented a novel candidate filtering heuristic to reduce the

model search space: we select as candidates only syntactic constituents that are *immediate* descendants of *all* S phrases that include the corresponding predicate. For both languages, over 99.6% of the candidates match this constraint. An additional filtering constraint implemented is that the model inspects only candidate labels allowed for the given predicate.[1] To enforce the domain constraints, i.e., no overlap allowed between arguments of the same predicate and numbered arguments (Arg0 to Arg5) can not repeat for the same predicate, we use a dynamic programming algorithm similar to the one proposed by Toutanova et al. [3].

The post-processing process in Model 1 recovers some temporal and locative modifier arguments missed during classification. The need for this component stemmed from the observation that in our initial experiments Model 1 performed poorly for the recognition of these two argument types on out-of-domain data. For example, simple constituents such as prepositional phrases starting with the preposition *durante* (*during*) were not recognized as temporal arguments even though samples existed in the training data. This happens because Model 1 focuses on other non-lexicalized features that appear to be more regular in the small training data available. To recover from this problem, we acquired a series of lexicalized patterns that can classify these modifier arguments with a precision higher than 60% in the training data.[2] The patterns have the following forms: (a) head word of candidate constituent, or (b) head word of constituent concatenated with the head word or POS tag of the first noun phrase in the constituent if the constituent is a prepositional phrase. During post-processing, these patterns are used to label constituents with temporal and locative argument labels only if they were not classified in any other class by the Model 1 classifiers.

3.2 Model 2

Unlike Model 1, Model 2 was developed from scratch for the two target languages. This model is an enhanced version of an earlier system [14] developed for the SemEval–2007 task *Multilevel Semantic Annotation of Catalan and Spanish* [15].

Candidate filtering in Model 2 has two significant differences from the strategy used in Model 1. First, Model 2 inspects only immediate descendants of the *most specific* S phrase that includes the target predicate. Second, the model skips constituents with the syntactic functions AO, ET, MOD, NEG, IMPERS, PASS, and VOC, as these never carry a semantic role in the training corpora. During training, these constituents are assigned an additional semantic label, NONE, which is not reported in the output. The intuition behind Model 2's candidate filtering heuristic is that, by filtering out constituents more aggressively than Model 1, this approach may miss some valid argument constituents but it generates a cleaner set of candidates with fewer negative samples.

For classification, Model 2 uses memory-based learning, specifically the IB1 algorithm as implemented in TiMBL[3]. IB1 is a supervised inductive algorithm for learning

[1] A verbal lexicon with this information is distributed with the corpora.

[2] This introduces some overfitting because we "know" that Model 1 performs poorly for these arguments on out-of-domain data. However, the overfitting is minimal because patterns are learned strictly from the training data. Using the post-processing component is important because it shows that local models can be successfully interleaved with global models.

[3] http://ilk.uvt.nl/timbl/

classification tasks based on the k-nearest neighbor classification rule. The algorithm is parametrized by using Jeffrey Divergence as the similarity metric, gain ratio for feature weighting, using 11 k-nearest neighbors, and weighting the class vote of neighbors as a function of their inverse linear distance. Similarly to Model 1, Model 2 uses features extracted from the argument and predicate constituents and the relation between the two phrases. Additionally, Model 2 extracts a series of features from the entire clause that includes the predicate in focus: total number of predicate siblings with function `CC`; relative positions of siblings with functions `SUJ`, `CAG`, `CD`, `CI`, `ATR`, `CPRED`, and `CREG` in relation to the verb; boolean feature that indicates if the clause contains a verbal *se*; and total number of predicate siblings in the clause. Model 2's complete feature set is detailed in [14].

Motivated by the observation that not all features have the same relevance, i.e., the most informative features for SRL in Spanish and Catalan are the features that provide information about the syntactic constituent of the candidate argument [16], Model 2 is the result of a feature selection process. Feature selection was performed by starting with a set of basic features (the head words with their POS tags, in their local context) and gradually adding new features. Every new feature added to the basic system was evaluated in terms of average accuracy in a 10-fold cross-validation experiment; if it improved the performance on held-out data, it was added to the selection.

Post-processing in Model 2 is a minimal process: it removes arguments tagged with the `NONE` label and constructs the required bracketing structure for the output. Because Model 2 extracts its candidate arguments only from sibling constituents we do not need to enforce the non-overlapping constraint. Enforcing the non-repetition constraint for numbered arguments did not improve results, so we did not include it in the final system.

3.3 Differences Between the Two Local Models

While both local models follow the same SRL approach, i.e., mapping each argument to one syntactic constituent and learning classifiers to assign valid argument labels to candidate constituents, there are significant differences between them: (a) The filtering strategy for candidate arguments is different: Model 2 uses an approach that generates fewer candidate arguments than Model 1. (b) Model 2 uses a richer feature set, e.g., it has features that exploit syntactic information from the whole clause that includes the predicate. Model 1 does not exploit clause-level information. (c) Model 2 performs feature selection whereas Model 1 does not. (d) The learning paradigm is different: AdaBoost versus memory-based learning. (e) Model 1 uses a series of post-processing patterns to recover some modifier arguments that are not captured during classification. These differences ensure that there is sufficient variance between the two local models, a crucial condition for the success of the combination model.

4 Global Models

4.1 The Re-ranking Model

We base our re-ranking approach on a variant of the re-ranking Perceptron of Collins and Duffy [17]. We modify the original algorithm in two ways to make it more robust to the small training set available: (a) instead of comparing the score of the correct frame

Algorithm 1: Re-ranking Perceptron

$\mathbf{w} = \mathbf{0}$
for $i = 1$ **to** n **do**
  for $j = 2$ **to** n_i **do**
    if $\mathbf{w} \cdot \mathbf{h}(\mathbf{x_{ij}}) > \mathbf{w} \cdot \mathbf{h}(\mathbf{x_{i1}}) - \tau$ **then**
      $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{h}(\mathbf{x_{i1}}) - \mathbf{h}(\mathbf{x_{ij}})$

only with that of the frame predicted by the current model, we sequentially compare it with the score of *each* candidate frame, and (b) we learn not only when the prediction is incorrect but also when the prediction is not confident enough. Both these changes allow the algorithm to acquire more information about the problem to be learned, an important advantage when the training data is scarce.

The algorithm is listed in Algorithm 1: $\mathbf{w}$ is the vector of model parameters, $\mathbf{h}$ generates the feature vector for one example, and $\mathbf{x_{ij}}$ denotes the jth candidate for the ith frame in the training data. $\mathbf{x_{i1}}$, which denotes the "correct" candidate for frame i, is selected in training to maximize the F_1 score for each frame. The algorithm sequentially inspects all candidates for each frame and learns when the difference between the scores of the correct and the current candidate is less than a threshold τ. During testing we use the average of all acquired model vectors, weighted by the number of iterations they survived in training. We tuned all system parameters through cross-validation on the training data. For both languages we set $\tau = 10$ (we do not normalize feature vectors) and the number of training epochs to 2.

With respect to the features used, we focus only on global features that can be extracted independently of the local models. We show in Section 5 that this approach performs better on the small corpora available than approaches that include features from the local models, which are too sparse when the learning sample is an entire frame. We group the features into two sets: (a) features that extract information from the whole candidate set, and (b) features that model the structure of each candidate frame:

Features from the whole candidate set:
(1) Position of the current candidate in the whole set. Frame candidates consistent with the domain constraints are generated using a dynamic programming algorithm [3], and then sorted in descending order of the log probability of the whole frame (i.e., the sum of all argument log probabilities as reported by the local model). Hence, smaller positions indicate candidates that the local model considers better.
(2) For each argument in the current frame, we store its number of repetitions in the whole candidate set. The intuition is that an argument that appears in many candidate frames is most likely correct.

Features from each candidate frame:
(3) The complete sequence of argument labels, extended with the predicate lemma and voice, similar to Toutanova et al. [3].
(4) Maximal overlap with a frame from the verb lexicon. Both the Spanish and Catalan TreeBanks contain a lexicon that lists the accepted sequences of arguments for the most common verbs. For each candidate frame, we measure the maximal overlap with the

lexicon frames for the given verb and use the precision, recall, and F_1 scores as features.
(5) Average probability (from the local model) of all arguments in the current frame.
(6) For each argument label that repeats in the current frame, we add combinations of the predicate lemma, voice, argument label, and the number of label repetitions as features. The intuition is that argument repetitions typically indicate an error (even if allowed by the domain constraints).

4.2 The Combination Model

The combination model is an adaptation of a global model we previously introduced for English [2]. The approach starts by merging the solutions generated by the two local models into a unique pool of candidate arguments, which are then re-scored using global information fed to a set of binary discriminative classifiers (one for each argument label). The classifiers assign to each argument a score measuring the confidence that the argument is part of the correct solution. Finally, the re-scored arguments for one sentence are merged into the best solution –i.e., the argument set with the highest combined score– that is consistent with the domain constraints. We implemented the discriminative classifiers using Support Vector Machines[4] configured with linear kernels with the default parameters. We implemented the solution generation stage with a CKY-based dynamic programming algorithm [18].

We group the features used by the re-scoring classifiers into four sets:

FS1. Voting features – these features quantify the votes received by each argument from the two local models. This feature set includes: (a) the *label* of the candidate argument; (b) the *number of systems* that generated an argument with this label and span; (c) the *unique ids* (Model 1 or Model 2) of the models that generated an argument with this label and span; and (d) the *argument sequence* of the whole frame for the models that generated this argument candidate.

FS2. Overlap features (same predicate) – these features measure the overlap between different arguments produced by the two models for the same predicate: (a) the *number* and *unique ids* of the models that generated an argument with the same span but different label; (b) the *number* and *unique ids* of the models that generated an argument that is included, or contains, or overlaps the candidate argument in focus.

FS3. Overlap features (other predicates) – these features are similar with the previous group, with the difference that we now compare arguments generated for *different* predicates. The motivation for the overlap features is that no overlap is allowed between arguments attached to the same predicate, and only inclusion or containment is permitted between arguments assigned to different predicates.

FS4. Features from the local models – we replicate the features from the local models that were shown to be the most effective for the SRL problem: information about the syntactic phrase and head word of the argument constituent (and left-most included noun phrase if the constituent is a prepositional phrase) and the syntactic path between the argument and predicate constituents. The motivation for these features is to learn the syntactic and lexical preferences of the individual models.

[4] `http://svmlight.joachims.org`

5 Experimental Results

For all the experiments, we used the corpora provided by the SRL task for Catalan (ca) and Spanish (es) at SemEval-2007 [7]. This is a part of the CESS-ECE corpus consisting of about 100K words per language, annotated with full parsing, syntactic functions, and semantic roles, and also including named entities and noun senses. The source of the corpus is varied including articles from news agencies, newspapers, and balanced corpora of the languages involved. These corpora are split into training (90%) and test (10%) subsets. Each test set is also divided into two subsets: 'in-domain' (marked with the .in suffix) and 'out-of-domain' (.out) test corpora. The first is intended to be homogeneous with respect to the training corpus and the second is extracted from a part of the CESS-ECE corpus not involved in the development of the resources.

Although the task at SemEval-2007 was to predict three layers of information, namely, semantic roles, named entities and noun senses, from the gold standard parse trees, we only address the SRL subtask in this work. It is worth noting that the role set used contains labels that are composed by a numbered argument (similar to PropBank) plus a verb-independent thematic role label similar to the scheme proposed in VerbNet.

The data for training the global models was generated by performing 5-fold cross validation on the whole training set with the previous models in the pipeline of processors (i.e., the individual models after post-processing). Also, parameter tuning was always performed by cross validation on the training set.

5.1 Overall Results

For the clarity of the exposition we present a top-down analysis of the proposed approach: we discuss first the results of the overall system and then we analyze the two global models in the system pipeline in the next two sub-sections.

We list the results of the complete system in the "Combination" rows in Table 1, for the four test corpora (Spanish and Catalan, in and out of domain). In this table we report results using the best configurations of the global models (see the next sub-sections for details). Next to the F_1 scores we list the corresponding statistical significance intervals obtained using bootstrap re-sampling [19].[5] The F_1 scores range from 83.56 (ca.out) to 88.88 (ca.in). These results are comparable to the best SRL systems for English, where the performance using correct syntactic information approaches 90 F_1 points for in-domain evaluation. We consider these numbers encouraging considering that our training corpora is 10 times smaller than the English PropBank and we have to label a larger number of classes (e.g., there are 33 core arguments for Spanish vs. 6 for English).

5.2 Analysis of the Combination Model

Table 1 details the contribution of the combination model, i.e., the right-most box in Figure 1. We compare the combination model against its two inputs, i.e., Models 1 and 2 after re-ranking and post-processing (position (c) in Figure 1), and against two baselines, one recall-oriented ("Baseline R") and one precision-oriented ("Baseline P").

[5] F_1 rates outside of these intervals are assumed to be significantly different from the related F_1 rate ($p < 0.05$).

	ca.in			ca.out		
	P	R	F_1	P	R	F_1
Combination	92.16	85.83	**88.88**$\pm$1.80	87.80	79.72	**83.56**$\pm$2.33
Model 1 (c)	87.83	83.61	87.22$\pm$2.10	82.83	79.25	81.00$\pm$2.71
Model 2 (c)	87.59	86.52	87.05$\pm$2.17	82.22	77.28	79.67$\pm$2.62
Baseline R	87.67	**87.22**	87.45$\pm$2.19	82.18	**82.25**	82.21$\pm$2.47
Baseline P	**94.30**	80.52	86.87$\pm$2.03	**92.11**	69.01	78.90$\pm$2.71
	es.in			es.out		
	P	R	F_1	P	R	F_1
Combination	89.22	81.09	**84.96**$\pm$1.80	89.75	83.46	**86.49**$\pm$1.79
Model 1 (c)	83.06	81.47	82.26$\pm$1.94	87.68	84.44	86.03$\pm$2.07
Model 2 (c)	83.42	82.47	82.94$\pm$2.10	85.41	85.41	85.41$\pm$1.89
Baseline R	82.88	**83.38**	83.13$\pm$2.02	84.92	**85.99**	85.45$\pm$1.78
Baseline P	**92.32**	73.66	81.94$\pm$2.15	**95.35**	77.82	85.70$\pm$1.79

Table 1. Overall results for the combination model. The individual models are evaluated after re-ranking and post-processing (where applicable), i.e., position (c) in Figure 1.

	ca.in			ca.out			es.in			es.out		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
FS1	92.28	84.17	88.04	88.54	71.83	79.32	**89.97**	79.71	84.53	90.62	81.81	85.99
+ FS2	92.22	85.57	88.77	87.76	79.44	83.39	88.64	**81.85**	**85.11**	89.54	**84.92**	**87.17**
+ FS3	92.16	**85.83**	**88.88**	87.80	**79.72**	**83.56**	89.22	81.09	84.96	89.75	83.46	86.49
+ FS4	**92.58**	84.61	88.41	**87.98**	77.00	82.12	89.73	79.63	84.38	**89.85**	81.81	85.64

Table 2. Feature analysis for the combination model.

Baseline R merges *all* the arguments generated by Models 1 and 2 (c). Baseline P selects only arguments where the two input models agreed.[6]

The combination model is better than its two inputs in all the setups. The increase in F_1 scores ranges from 0.46 points (es.out) to 2.56 points (ca.out). For in-domain data (ca.in and es.in), the F_1 score improvement is approximately 2 points, which is similar to the improvements seen for English, even though here we have less training data and fewer individual models that provide less information (e.g., no output probabilities are available). The performance of the combination model is always better than both of the baselines as well. As expected, the recall-oriented baseline achieves the highest recall and the precision-oriented baseline the highest precision, but the combination model obtains the best F_1 score. This is an indication that the model is capable of learning useful information beyond the simple redundancy used by the baselines.

Table 2 analyzes the contribution of the four proposed features groups. The analysis is cumulative, i.e., the "+ FS2" row lists the performance of the system configured with the first two feature groups. The table indicates that in our setup the features from the local models (FS4) do not help. This is a significant difference from English SRL, where lexical and syntactic features extracted from the local models are known to help global strategies [2, 3]. Our conjecture is that in our setup the combination model can not recover from the increased sparsity introduced by the features that model syntactic context and lexical information. Note that the sparsity of these features is much larger

[6] Conflicts with the domain constraints are solved using the same strategy as [2].

	ca.in			ca.out			es.in			es.out		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Model 1 (a)	85.48	84.43	84.95	80.73	68.45	74.09	78.73	77.11	77.91	84.73	73.93	78.96
This paper	87.83	86.61	87.22	**82.17**	**69.67**	**75.41**	**83.06**	**81.47**	**82.26**	**86.63**	**76.26**	**81.12**
Collins	**87.92**	**86.70**	**87.30**	81.19	68.92	74.56	82.67	81.09	81.87	85.62	75.88	80.45
Toutanova	79.40	78.43	78.92	73.00	62.44	67.31	79.32	76.95	78.12	82.52	75.29	78.74

Table 3. Analysis of the re-ranking model.

in the combination model than the local models because at this level we work only with the final output of the local models, whereas the individual models have a much larger space of candidate arguments. A somewhat similar observation can be made for FS3. But because we performed the tuning of the combination model on training data and there we saw a small improvement when using this feature group we decided to include this set of features in the best model configuration.

5.3 Analysis of Re-ranking

We analyze the proposed re-ranking model in Table 3. We compare the re-ranking performance against the corresponding local model (Model 1 (a)) and against two variations of our approach: in the first we used our best feature set but the original re-ranking Perceptron of Collins and Duffy [17], and in the second we used our re-ranking algorithm but we configured it with the features proposed by Toutanova et al. [3]. This feature set includes features (3) and (6) from Section 4.1 and all features from the local model concatenated with the label of the corresponding candidate argument.

We draw several observations from this analysis: (a) our re-ranking model always outperforms the local model, with F_1 score improvements ranging from 1.32 to 4.35 points; (b) the re-ranking Perceptron proposed here performs better than the algorithm of Collins and Duffy on three out of four corpora, and (c) the feature set proposed here achieve significant better performance on the SemEval corpora than the set proposed by Toutanova et al., which never improves over the local model. These observations indicate that, while our modifications to the re-ranking Perceptron yield a minor improvement, the biggest contribution comes from the novel set of features proposed. Whereas the model configured with the Toutanova et al. feature set performs modestly because the features from the local models are too sparse in this global setup, we replicate the behavior of the local model just with feature (1), and all the other five global features proposed have a positive contribution. This conclusion correlates well with the analysis in the previous sub-section, where we also observed that syntactic and lexical features from the local model do not help in another global setup with small training corpora.

5.4 Putting It All Together: Analysis of Stacking

Table 4 summarizes the paper's main results. We show the results at every point in the pipeline for Model 1: after the local model, after re-ranking, after post-processing, and after the combination with Model 2. Note that we apply the post-processing patterns only on out-of-domain data because this is where we observed that the local model fails to recognize locative and temporal modifier arguments. The table re-enforces our claim that the stacking of global strategies is a successful way to mitigate the lack of training data, even (or more so) when the global models are interleaved with local strategies.

	ca.in			ca.out		
	P	R	F_1	P	R	F_1
Model 1 (a)	85.48	84.43	84.95$\pm$2.27	80.73	68.45	74.09$\pm$3.18
+ re-ranking	87.83	**86.61**	87.22$\pm$2.00	82.17	69.67	75.41$\pm$2.95
+ post-processing	–	–	–	82.83	79.25	81.00$\pm$3.16
+ combination	**92.16**	85.83	**88.88**$\pm$1.80	**87.80**	**79.72**	**83.56**$\pm$2.33
	es.in			es.out		
	P	R	F_1	P	R	F_1
Model 1 (a)	78.73	77.11	77.91$\pm$2.27	84.73	73.93	78.96$\pm$2.50
+ re-ranking	83.06	**81.47**	82.26$\pm$2.04	86.63	76.26	81.12$\pm$2.21
+ post-processing	–	–	–	87.68	84.44	86.03$\pm$2.23
+ combination	**89.22**	81.09	**84.96**$\pm$1.80	**89.75**	**83.46**	**86.49**$\pm$1.79

Table 4. Stacking results relative to Model 1.

On in-domain corpora (ca.in and es.in) we improve the performance of the local model with 3.93 and 7.05 F_1 points. On out-of-domain corpora (ca.out and es.out), where we applied the post-processing patterns, we increased the F_1 score of the local model with 9.47 and 7.53 points.

Another important observation is that the two global approaches can be stacked because they provide complementary benefits. Because re-ranking is configured to optimize F_1 it tends to improve recall, which is generally lower in the local model due to the insufficient coverage of the training data. On the other hand, the combination model tends to improve precision because a good part of its learning is driven by the redundancy between the two models, which is a precision-oriented feature.

6 Conclusions

In this paper we propose a SRL approach that stacks two joint (or global) inference components on top of two individual (or local) SRL models. The first global model re-ranks entire candidate frames produced by the local models. The second joint inference model combines the outputs of the two local models after re-ranking using meta-learning and sentence-level information.

We draw several novel conclusions from this work. First, we show that global strategies work well under unfavorable training conditions, e.g., our training corpora are 10 times smaller than the English PropBank, there are only two local models available for combination, and these models provide limited information (no output probabilities). We show that a key requirement for success in these conditions is to focus on global mostly-unlexicalized features that have low sparsity even in small training corpora. We propose such feature sets both for the re-ranking and the combination models. We also show that lexical and syntactic features from the local models, which tend to have high sparsity, do not help in our setup. A second novelty of our work is that we show that the proposed global strategies can be successfully stacked because they provide complementary benefits: in our configuration re-ranking tends to improve recall whereas the combination model boosts precision.

Our complete SRL system obtains the current best results for Spanish and Catalan: for in-domain data and correct syntactic information, our system obtains F_1 scores of 88.88 points for Catalan and 84.96 for Spanish. For out-of-domain data and gold syntax, our systems obtains 83.56 points for Catalan and 86.49 for Spanish.

Acknowledgements

Mihai Surdeanu is a research fellow within the Ramón y Cajal program of the Spanish Ministry of Science and Technology. Roser Morante was funded by the postdoctoral grant EX2005–1145 awarded by the latter institution to the project *Técnicas semiautomáticas para el etiquetado de roles semánticos en corpus del español*. This research has been partially funded by the Spanish Ministry of Education and Science, projects OpenMT (TIN2006-15307-C03-02) and TRANGRAM (TIN2004-07925-C03-02).

References

1. Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J., Jurafsky, D.: Support vector learning for semantic argument classification. Machine Learning **60** (2005) 11–39
2. Surdeanu, M., Màrquez, L., Carreras, X., Comas, P.R.: Combination strategies for semantic role labeling. Journal of Artificial Intelligence Research (JAIR) **29** (2007) 105–151
3. Toutanova, K., Haghighi, A., Manning, C.D.: A global joint model for semantic role labeling. Computational Linguistics **33** (2008) forthcoming
4. Punyakanok, V., Roth, D., Yih, W.: The importance of syntactic parsing and inference in semantic role labeling. Computational Linguistics **33** (2008) forthcoming
5. Pradhan, S., Hacioglu, K., Ward, W., Martin, J.H., Jurafsky, D.: Semantic role chunking combining complementary syntactic views. In: Proc. of CoNLL-2005. (2005)
6. Xue, N.: Labeling chinese predicates with semantic roles. Computational Linguistics **33** (2008) forthcoming
7. Màrquez, L., Villarejo, L., Martí, M., Taulé, M.: Semeval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In: Proc. of SemEval-2007. (2007)
8. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning **37** (1999)
9. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics **28** (2002) 245–288
10. Surdeanu, M., Harabagiu, S., Williams, J., Aarseth, P.: Using predicate arguments structures for information extraction. In: Proc. of ACL. (2003)
11. Xue, N., Palmer, M.: Calibrating features for semantic role labeling. In: Proc. of EMNLP. (2004)
12. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 shared task: Semantic role labeling. In: Proc. of CoNLL–2005. (2005)
13. Màrquez, L., Comas, P., Giménez, J., Català, N.: Semantic role labeling as sequential tagging. In: Proc. of CoNLL–2005. (2005)
14. Morante, R., Busser, B.: ILK2: Semantic role labelling for Catalan and Spanish using TiMBL. In: Proc. of SemEval-2007. (2007)
15. Màrquez, L., Villarejo, L., Martí, M., Taulé, M.: Semeval-2007 task 09: Multilevel semantic annotation of Catalan and Spanish. In: Proc. of SemEval-2007. (2007)
16. Morante, R., van den Bosch, A.: Memory–based semantic role labelling. In: Proc. of RANLP. (2007)
17. Collins, M., Duffy, N.: New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In: Proc. of ACL. (2002)
18. Younger, D.H.: Recognition and parsing of context-free languages in n^3 time. Information and Control **10** (1967) 189–208
19. Noreen, E.W.: Computer-Intensive Methods for Testing Hypotheses. John Wiley & Sons (1989)