Generating High Quality Proposition Banks for Multilingual Semantic Role Labeling

Alan Akbik∗
Technische Universität Berlin, Germany
alan.akbik@tu-berlin.de
Laura Chiticariu Marina Danilevsky Yunyao Li
Shivakumar Vaithyanathan Huaiyu Zhu
IBM Research - Almaden
650 Harry Road, San Jose, CA 95120, USA
{chiti,mdanile,yunyaoli,vaithyan,huaiyu}@us.ibm.com

Abstract
Semantic role labeling (SRL) is crucial to natural language understanding as it identifies the predicate-argument structure in text with semantic labels. Unfortunately, resources required to construct SRL models are expensive to obtain and simply do not exist for most languages. In this paper, we present a two-stage method to enable the construction of SRL models for resource-poor languages by exploiting monolingual SRL and multilingual parallel data. Experimental results show that our method outperforms existing methods. We use our method to generate Proposition Banks with high to reasonable quality for 7 languages in three language families and release these resources to the research community.

1 Introduction
Semantic role labeling (SRL) is the task of automatically labeling predicates and arguments in a sentence with shallow semantic labels. This level of analysis provides a more stable semantic representation across syntactically different sentences, thereby enabling a range of NLP tasks such as information extraction and question answering (Shen and Lapata, 2007; Maqsud et al., 2014). Projects such as the Proposition Bank (PropBank) (Palmer et al., 2005) spent considerable effort to annotate corpora with semantic labels, in turn enabling supervised learning of statistical SRL parsers for English. Unfortunately, due to the high costs of manual annotation, comparable SRL resources do not exist for most other languages, with few exceptions (Hajić et al., 2009; Erk et al., 2003; Zaghouani et al., 2010; Vaidya et al., 2011).

As a cost-effective alternative to manual annotation, previous work has investigated the direct projection of semantic labels from a resource rich language (English) to a resource poor target language (TL) in parallel corpora (Pado, 2007; Van der Plas et al., 2011). The underlying assumption is that original and translated sentences in parallel corpora are semantically broadly equivalent. Hence, if English sentences of a parallel corpus are automatically labeled using an SRL system, these labels can be projected onto aligned words in the TL corpus, thereby automatically labeling the TL corpus with semantic labels. This way, PropBank-like resources can automatically be created that enable the training of statistical SRL systems for new TLs.

However, as noted in previous work (Pado, 2007; Van der Plas et al., 2011), aligned sentences in parallel corpora often exhibit issues such as translation

![Figure 1: Pair of parallel sentences from French with word alignments (dotted lines), SRL labels for the English sentence, and gold SRL labels for the French sentence. Only two of the seven English SRL labels should be projected here.](image-url)
shifts that go against this assumption. For example, in Fig. 1, the English sentence “We need to hold people responsible” is translated into a French sentence that literally reads as “There need to exist those responsible”. Hence, the predicate label of the English word “hold” should not be projected onto the French verb, which has a different meaning. As the example in Fig. 1 shows, this means that only a subset of all SL labels can be directly projected.

In this paper, we aim to create PropBank-like resources for a range of languages from different language groups. To this end, we propose a two-stage approach to cross-lingual semantic labeling that addresses such errors, shown in Fig. 2. Given a parallel corpus in which the source language (SL) side is automatically labeled with PropBank labels and the TL side is syntactically parsed, we use a filtered projection approach that allows the projection only of high-confidence SL labels. This results in a TL corpus with low recall but high precision. In the second stage, we repeatedly sample a subset of complete TL sentences and train a classifier to iteratively add new labels, significantly increasing the recall in the TL corpus while retaining the improvement in precision.

Our contributions are: (1) We propose filtered projection focused specifically on raising the precision of projected labels, based on a detailed analysis of direct projection errors. (2) We propose a bootstrap learning approach to retrain the SRL to iteratively improve recall without a significant reduction of precision, especially for arguments; (3) We demonstrate the effectiveness and generalizability of our approach via an extensive set of experiments over 7 different language pairs. (4) We generate PropBanks for each of these languages and release them to the research community.

2 Stage 1: Filtered Annotation Projection

Stage 1 of our approach (Fig. 2) is designed to create a TL corpus with high precision semantic labels.

Direct Projection The idea of direct annotation projection (Van der Plas et al., 2011) is to transfer semantic labels from SL sentences to TL sentences according to word alignments. Formally, for each pair of sentences $s_{SL}$ and $s_{TL}$ in the parallel corpus, the word alignment produces alignment pairs $(w_{SL,i}, w_{TL,i'})$, where $w_{SL,i}$ and $w_{TL,i'}$ are words from $s_{SL}$ and $s_{TL}$ respectively. Under direct projection, if $l_{SL,i}$ is a predicate label for $w_{SL,i}$ and $(w_{SL,i}, w_{TL,i'})$ is an alignment pair, then $l_{SL,i}$ is transferred to $w_{TL,i'}$; If $l_{SL,j}$ is a predicate-argument label for $(w_{SL,i}, w_{SL,j})$, and $(w_{SL,i}, w_{TL,i'})$ and $(w_{SL,j}, w_{TL,j'})$ are alignment pairs, then $l_{SL,j}$ is transferred to $(w_{TL,i'}, w_{TL,j'})$, as illustrated below.

Filtered Projection As discussed earlier, direct projection is vulnerable to errors stemming from issues such as translation shifts. We propose filtered projection focused specifically on improving the precision of projected labels. Specifically, for a pair of sentences $s_{SL}$ and $s_{TL}$ in the parallel corpus, we retain the semantic label $l_{SL,i}$ projected from $w_{SL,i}$ onto $w_{TL,i'}$ if and only if it satisfies the filtering policies. This results in a target corpus containing fewer labels but of higher precision compared to that obtained via direct projection.

In the rest of the section, we analyze typical errors in direct projection (Sec. 2.2), present a set of filters to handle such errors (Sec. 2.3), and experimentally evaluate their effectiveness (Sec. 2.4).

1 The resources are available on request.
### Table 1: Breakdown of error classes in predicate projection.

<table>
<thead>
<tr>
<th>Error Class</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation Shift: Predicate Mismatch</td>
<td>37</td>
</tr>
<tr>
<td>Translation Shift: Verb→Non-verb</td>
<td>36</td>
</tr>
<tr>
<td>No English Equivalent</td>
<td>8</td>
</tr>
<tr>
<td>Gold Data Errors</td>
<td>6</td>
</tr>
<tr>
<td>SRL Errors</td>
<td>5</td>
</tr>
<tr>
<td>Verb (near-)Synonyms</td>
<td>4</td>
</tr>
<tr>
<td>Light Verb Construction</td>
<td>3</td>
</tr>
<tr>
<td>Alignment Errors</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>100</td>
</tr>
</tbody>
</table>

### Table 2: Breakdown of error classes in argument projection.

<table>
<thead>
<tr>
<th>Error Class</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Argument Head</td>
<td>33</td>
</tr>
<tr>
<td>SRL Errors</td>
<td>31</td>
</tr>
<tr>
<td>No English Equivalent</td>
<td>12</td>
</tr>
<tr>
<td>Gold Data Errors</td>
<td>11</td>
</tr>
<tr>
<td>Translation Shift: Argument Function</td>
<td>6</td>
</tr>
<tr>
<td>Parsing Errors</td>
<td>4</td>
</tr>
<tr>
<td>Alignment Errors</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>100</td>
</tr>
</tbody>
</table>

2.1 Experimental Setup

**Data** For experiments in this section and Sec. 3, we used the gold data set compiled by (Van der Plas et al., 2011), referred to as French$_{gpl}$. It consists of 1,000 sentence-pairs from the English-French Europarl corpus (Koehn, 2005) with French sentences manually labeled with predicate and argument labels from the English Propbank.

**Evaluation** In line with previous work (Van der Plas et al., 2010), we count synonymous predicate labels sharing the same VERBNET (Schuler, 2005) class as true positives. In addition, we exclude modal verbs from the evaluation due to inconsistent annotation.

**Source Language SRL** Throughout the rest of the paper, we use CLEARNLP (Choi and McCallum, 2013), a state-of-the-art SRL system, to produce semantic labels for English text.

2.2 Error Analysis

We observe that direct projection labels have both low precision and low recall (see Tab.3 (Direct)).

**Analysis of False Negatives** The low recall of direct projection is not surprising; most semantic labels in the French sentences do not appear in the corresponding English sentences at all. Specifically, among 1,741 predicate labels in the French sentences, only 778 exist in the corresponding English sentences, imposing a 45% upper bound on the recall for projected predicates. Similarly, of the 5,061 argument labels in the French sentences, only 1,757 exist in the corresponding English sentences, resulting in a 35% upper bound on recall for arguments.

**Analysis of False Positives** While the recall produced by direct projection is close to the theoretical upper bound, the precision is far from the theoretical upper bound of 100%. To understand causes of false positives, we examine a random sample of 200 false positives, of which 100 are incorrect predicate labels, and 100 are incorrect argument labels belonging to correctly projected predicates. Tab.1 and 2 show the detailed breakdown of errors for predicates and arguments, respectively. We first analyze the most common types of errors and discuss the residual errors later in Sec. 2.5.

- **Translation Shift: Predicate Mismatch** The most common predicate errors (37%) are translation shifts in which an English predicate is aligned to a French verb with a different meaning. Fig. 1 illustrates such a translation shift: label **hold.01** of English verb **hold** is wrongly projected onto the French verb **ait**, which is labeled as **exist.01** in French$_{gpl}$.

- **Translation Shift: Verb→Non-Verb** is another common predicate error (36%). English verbs may be aligned with TL words other than verbs, which is often indicative of translation shifts. For instance, in the following sentence pair
  
  _su_ We know what happened
  _fr_ On connait la suite

  the English verb **happen** is aligned to the French noun **suite** (*result*), causing it to be wrongly projected with the English predicate label **happen.01**.

- **Non-Argument Head** The most common argument error (33%) is caused by the projection of argument labels onto words other than the syntactic head of a target verb’s argument. For example, in Fig. 1 the label **A1** on the English **hold** is wrongly transferred to the French **ait**, which is not the syntactic head of the complement.
2.3 Filters
We consider the following filters to remove the most common types of false positives.

**Verb Filter (VF)** targets Verb→Non-Verb translation shift errors (Van der Plas et al., 2011). Formally, if direct projection transfers predicate label \(l_{a,i}\) from \(w_{s,i}\) onto \(w_{t,i'}\), retain \(l_{a,i}\) only if both \(w_{s,i}\) and \(w_{t,i'}\) are verbs.

**Translation Filter (TF)** handles both Predicate Mismatch and Verb→Non-Verb translation shift errors. It makes use of a translation dictionary and allows projection only if the TL verb is a valid translation of the SL verb. In addition, in order to ensure consistent predicate labels throughout the TL corpus, if a SL verb has several possible synonymous translations, it allows projection only for the most commonly observed translation.

Formally, for an aligned pair \((w_{s,i}, w_{t,i'})\) where \(w_{s,j}\) has predicate label \(l_{a,i}\), if \((w_{s,i}, w_{t,j'})\) is not a verb to verb translation from SL to TL, assign no label to \(w_{t,i'}\). Otherwise, split the set of SL translations of \(w_{t,i'}\) into synonym sets \(S_1, S_2, \ldots\); For each \(S_k\), let \(W^k\) be the subset of \(S_k\) most commonly aligned with \(w_{t,i'}\); If \(w_{s,i}\) is in one of these \(W^k\), assign label \(l_{a,i}\) to \(w_{t,i'}\); Otherwise assign no label to \(w_{t,i'}\).

**Reattachment Heuristic (RH)** targets non-argument head errors that occur if a TL argument is not the direct child of a verb in the dependency parse tree of its sentence. \(^4\)Assume direct projection transfers the predicate-argument label \(l_{s,j}\) from \((w_{s,i}, w_{s,j})\) onto \((w_{t,i'}, w_{t,j'})\). Find the immediate ancestor verb of \(w_{t,j'}\) in the dependency parse tree. Denote as \(w_{t,k}\) its child that is an ancestor of \(w_{t,i'}\). Assign the label \(l_{s,j}\) to \((w_{t,i'}, w_{t,k})\) instead of \((w_{t,i'}, w_{t,j'})\). An illustration is below:

\[\text{RH ensures that labels are always attached to the syntactic heads of their respective arguments, as defined over constituent-based trees to reduce the set of viable nodes for argument labels to all nodes that are not a child of some ancestor of the predicate.}\]

\[\text{In [Padó and Lapata, 2009], a similar filtering method is discussed in Sec. 2.2. Most of the remaining errors} \]

\[\text{Table 3: Quality of predicate and argument labels for different projection methods on French_{test}, including upper bound.}\]

<table>
<thead>
<tr>
<th>Projection</th>
<th>Predicate</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Direct</td>
<td>0.45</td>
<td>0.43</td>
</tr>
<tr>
<td>VF</td>
<td>0.59</td>
<td>0.48</td>
</tr>
<tr>
<td>TF</td>
<td><strong>0.88</strong></td>
<td>0.36</td>
</tr>
<tr>
<td>VF+RH</td>
<td>0.59</td>
<td>0.48</td>
</tr>
<tr>
<td>TF+RH</td>
<td><strong>0.88</strong></td>
<td>0.36</td>
</tr>
<tr>
<td>Upper Bound</td>
<td>1</td>
<td>0.45</td>
</tr>
</tbody>
</table>

\[\text{VF reduces the number of wrongly projected predicate labels, resulting in an increase of predicate precision to 59\% (↑14 pp), without impact to recall. As a side effect, argument precision also increases to 53\% (↑10 pp), since, if a predicate label cannot be projected, none of its arguments can be projected. TF reduces the number of wrongly projected predicate labels even more significantly, increasing predicate precision to 88\% (↑43 pp), at a small cost to recall. Again, argument precision increases as a side effect. However, as expected, argument recall decreases significantly (↓14 pp, to 17\%), as many arguments can no longer be projected. RH targets argument labels directly (unlike VF and TF), significantly increasing argument precision and slightly increasing argument recall.}\]

In summary, initial experiments confirm that our proposed filters are effective in improving precision of projected labels at a small cost in recall. In fact, TF+RH results in nearly 100\% improvement in predicate and argument labels precision with a much smaller drop in recall.

2.5 Residual Errors
Filtered projection removes the most common errors discussed in Sec. 2.2. Most of the remaining errors
come from the following sources.

**SRL Errors** The most common residual errors in the remaining projected labels, especially for argument labels, are caused by mistakes made by the English SRL system. Any wrong label it assigns to an English sentence may be projected onto the TL sentence, resulting in false positives.

**No English Equivalent** A small number of errors occur due to French particularities that do not exist in English. Such errors include certain French verbs for which no appropriate English PropBank labels exist, and French-specific syntactic particularities.

**Gold Data Errors** Our evaluation so far relies on French gold as ground truth. Unfortunately, French gold does contain a small number of errors (e.g., missing argument labels). As a result, some correctly projected labels are being mistaken as false positives, causing a drop in both precision and recall. We therefore expect the true precision and recall of the approach to be somewhat higher than the estimate based on French gold.

### 3 Stage 2: Bootstrapped Training of SRL

As discussed earlier, the TL corpus generated via filtered projection suffers from low recall. We address this issue with the second stage of our method.

**Relabeling** The idea of relabeling (Van der Plas et al., 2011) is to first train an SRL system over a TL corpus labeled using direct projection (with VF filter) and then use this SRL to relabel the corpus, effectively overwriting the projected labels with potentially less noisy predicted labels.

We first present an analysis on relabeling in concert with our proposed filters (Sec. 3.1), which motivates our bootstrap algorithm (Sec. 3.2).

#### 3.1 Analysis of Relabeling Approach

We use the same experimental setup as in Sec. 2 and produce a labeled French corpus for each filtered annotation method. We then train an off-the-shelf SRL system (Björkelund et al., 2009) on each generated corpus and use it to relabel the corpus.

We measure precision and recall of each resulting TL corpus against French gold (see Tab. 4). Across all experiments, relabeling consistently improves recall over projection. The results also show how different factors affect the performance of relabeling.

#### Supplement vs. Overwrite Projected Labels

The labels produced by the trained SRL can be used to either overwrite projected labels as in (Van der Plas et al., 2011), or to supplement them (supplying labels only for words w/o projected labels). Whether to overwrite or supplement depends on whether labels produced by the trained SRL are of higher quality than the projected labels. We find that while predicted labels are of higher precision than directly projected labels, they are of lower precision than labels post filtered projection. Therefore, for filtered projection, it makes more sense to allow predicted labels to only supplement projected labels.

#### Impact of Sampling Method

We are further interested in learning the impact of sampling the data on the quality of relabeling. For the best filter found earlier (TF+RH), we compare SRL trained on the entire data set (full data) with SRL trained only on the subset of completely annotated sentences (comp. sent.), where completeness is defined as:

**Definition 1.** A direct component of a labeled sentence $s_{TL}$ is either a verb in $s_{TL}$ or a syntactic dependent of a verb. Then $s_{TL}$ is $k$-complete if $s_{TL}$ contains equal to or fewer than $k$ unlabeled direct compo-
provides precision. While this finding may seem counterintuitive, it can be explained by how statistical SRL works. A densely labeled training data set of labels \( L_\text{comp} \) increases monotonically in each iteration of Algorithm 1.

**Algorithm 1** Bootstrap learning algorithm

**Require:** Corpus \( C_n \) with initial set of labels \( L_n \), and resampling threshold function \( k(i) \);

for \( i = 1 \) to \( \infty \) do

Let \( k_i = k(i) \);

Let \( C_n^\text{comp} = \{ w \in C_n : w \in s_n, s_n \text{ is } k_i \text{-complete} \} \);

Let \( L_n^\text{comp} \) be subset of \( L_n \) appearing on \( C_n^\text{comp} \);

Train an SRL on \( (C_n^\text{comp}, L_n^\text{comp}) \);

Use the SRL to produce label set \( L_n^\text{new} \) on \( C_n^\text{new} \);

Let \( C_n^\text{suppl} = \{ w \in C_n : w \text{ not labelled by } L_n \} \);

Let \( L_n^\text{suppl} \) be subset of \( L_n^\text{new} \) appearing on \( C_n^\text{suppl} \);

if \( L_n^\text{suppl} = \emptyset \) then

Return the SRL;

end if

Let \( L_n = L_n \cup L_n^\text{suppl} \);

end for

nts. 0-complete is abbreviated as complete.

We observe that for TF+RH, when new labels supplement projected labels, relabeling over complete sentences results in better recall at slightly reduced precision, while including incomplete sentences into the training data reduces recall, but improves precision. While this finding may seem counterintuitive, it can be explained by how statistical SRL works. A densely labeled training data (such as \( \text{comp} \) subset) usually results in an SRL that generates densely labeled sentences, resulting in better recall but poorer precision. On the other hand, training data that is sparsely labeled results in an SRL that weighs the option of not assigning a label with higher probability, resulting in better precision and poorer recall. In short, one can control the trade-off between precision and recall of SRL output by manipulating the completeness of the training data.

### 3.2 Bootstrap Learning

Building on the observation that we can sample data in such a way as to either favor precision or recall, we propose a bootstrapping algorithm to train an SRL iteratively over \( k\)-complete subsets of the data which are supplemented by high precision labels produced from previous iteration. The detailed algorithm is depicted in Algorithm 1.

**Resampling Threshold** Our goal is to use bootstrap learning to improve recall without sacrificing too much precision.

**Proposition 1.** Under any resampling threshold, the set of labels \( L_n \) increases monotonically in each iteration of Algorithm 1.

Since Prop. 1 guarantees the increase of the set of labels, we need to select a resampling function to favor precision while improving recall. Specifically, we use the formula \( k(i) = \max(k_0 - i, 0) \), where \( k_0 \) is sufficiently large. Since the precision of labels generated by the SRL is lower than the precision of labels obtained from filtered projection, the precision of the training data is expected to decrease with the increase in recall. Therefore, starting with a high \( k \) seeks to ensure high precision labels are added to the training data in the first iterations. Decreasing \( k \) in each iteration seeks to ensure that resampling is done in an increasingly restrictive way to ensure that only high-quality annotated sentences are added to the training data, thus maintaining a high confidence in the learned SRL model.

### 3.3 Effectiveness of Bootstrapping

We experimentally evaluate the effectiveness of our model with \( k_0 = 9 \). As shown in Tab 4, bootstrapping outperforms relabeling, producing labels with best overall quality in terms of \( F_1 \) measure and recall for both predicates and arguments, with a relatively small cost in precision.

While Algorithm 1 guarantees the increase of recall (Prop. 1), it provides no such guarantee on precision. Therefore, it is important to experimentally decide an early termination cutoff before the SRL gets overtrained. To do so, we evaluated the performance of the bootstrapping algorithm at each iteration (Fig. 3). We observe that for the first 3 iterations, \( F_1 \)-measure for both predicates and arguments rises due to large increase in recall which offsets the smaller drop in precision. Then \( F_1 \)-measure remains stable, with recall rising and pre-
cision falling slightly at each iteration until convergence. To optimize precision and avoid overtraining, we set an iteration cutoff of 3. This combination of TF+RH filters, bootstrapping with $k_0 = 9$ and an iteration cutoff of 3 is used in the rest of our evaluation (Sec. 4), denoted as $FB_{best}$.

4 Multilingual Experiments

We use our method to generate Proposition Banks for 7 languages and evaluate the generated resources. We seek to answer the following questions: (1) What is the estimated quality for the generated PropBanks? How well does the approach work without language-specific adaptation? (2) Are there notable differences in quality from language to language; if so, why? We also present initial investigations on how different factors affect the performance of our method.

4.1 Experimental Setup

Data Tab. 5 lists the 7 different TLs and resources used in our experiments.\(^8\) We chose these TLs because (1) they are among top 10 most influential languages in the world (Weber, 1997); and (2) we could find language experts to evaluate the results. English is used as SL in all our experiments.

Approach Tested For each TL, we used $FB_{best}$ (Sec. 3.3) to generate a corpus with semantic labels. From each TL corpus, we extracted all complete sentences to form the generated PropBanks.

\(^8\)From each parallel corpus, we only keep sentences that are considered well-formed based on a set of standard heuristics. For example, we require a well-formed sentence to end in punctuation and not to contain certain special characters. For Arabic, as the dependency parser we use has relatively poor parsing accuracy, we additionally require sentences to be shorter than 100 characters.

Manual Evaluation While a gold annotated corpus for French ($\text{French}_G$) was available for our experiments in the previous Sections, no such resources existed for the other TLs we wished to evaluate. We therefore chose to conduct a manual evaluation for each TL, each executed identically: For each TL we randomly selected 100 complete sentences with their generated semantic labels and assigned them to two language experts who were instructed to evaluate the semantic labels (based on their English descriptions) for the predicates and their core arguments. For each label, they were asked to determine (1) whether the label is correct; (2) if yes, then whether the boundary of the labeled constituent is correct: If also yes, mark the label as fully correct, otherwise as partially correct.

Metrics We used the standard measures of precision, recall, and F1 to measure the performance of the SRLs, with the following two schemes: (1) Exact: Only fully correct labels are considered as true positives; (2) Partial: Both fully and partially correct matches are considered as true positives.\(^9\)

4.2 Experimental Results

Tab. 6 summarizes the estimated quality of semantic labels generated by our method for all seven TL. As can be seen, our method performed well for all

\(^9\)Note that since the manually evaluated semantic labels are only a small fraction of the labels generated, the performance numbers obtained from manual evaluation is only an estimate of the actual quality for the generated resources. Thus the numbers obtained based on manual evaluation cannot be directly compared against the numbers computed over $\text{French}_G$.

---

<table>
<thead>
<tr>
<th>Language</th>
<th>Dep. Parser</th>
<th>Data Set</th>
<th>#Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>STANFORD</td>
<td>UN</td>
<td>481K</td>
</tr>
<tr>
<td>Chinese</td>
<td>MATE-G</td>
<td>UN</td>
<td>2,986K</td>
</tr>
<tr>
<td>French</td>
<td>MATE-T</td>
<td>UN</td>
<td>2,542K</td>
</tr>
<tr>
<td>German</td>
<td>MATE-T</td>
<td>Europarl</td>
<td>560K</td>
</tr>
<tr>
<td>Hindi</td>
<td>MALT</td>
<td>Hindencorp</td>
<td>54K</td>
</tr>
<tr>
<td>Russian</td>
<td>MALT</td>
<td>UN</td>
<td>2,638K</td>
</tr>
<tr>
<td>Spanish</td>
<td>MATE-G</td>
<td>UN</td>
<td>2,304K</td>
</tr>
</tbody>
</table>

Table 5: Experimental setup

<table>
<thead>
<tr>
<th>ISO Code</th>
<th>Name</th>
<th>Language</th>
<th>Exact Match</th>
<th>Exact Recall</th>
<th>Exact F1</th>
<th>Partial Match</th>
<th>Partial Recall</th>
<th>Partial F1</th>
<th>Avg. (\kappa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>part.</td>
<td>0.97 0.89 0.83</td>
<td>0.69 0.77 0.72</td>
<td>0.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.97 0.93 0.67</td>
<td>0.63 0.65 0.85</td>
<td>0.77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>part.</td>
<td>0.97 0.88 0.93</td>
<td>0.83 0.95 0.91</td>
<td>0.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.97 0.92 0.82</td>
<td>0.81 0.92 0.86</td>
<td>0.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>French</td>
<td>part.</td>
<td>0.95 0.92 0.94</td>
<td>0.76 0.97 0.95</td>
<td>0.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.95 0.92 0.86</td>
<td>0.74 0.8 0.95</td>
<td>0.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>German</td>
<td>part.</td>
<td>0.96 0.92 0.94</td>
<td>0.73 0.83 0.95</td>
<td>0.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.96 0.92 0.91</td>
<td>0.73 0.81 0.92</td>
<td>0.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hindi</td>
<td>part.</td>
<td>0.91 0.68 0.78</td>
<td>0.93 0.94 0.98</td>
<td>0.88</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.91 0.68 0.78</td>
<td>0.54 0.68 0.81</td>
<td>0.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Russian</td>
<td>part.</td>
<td>0.96 0.94 0.95</td>
<td>0.79 0.97 0.94</td>
<td>0.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.96 0.94 0.79</td>
<td>0.65 0.72 0.93</td>
<td>0.89</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td>part.</td>
<td>0.96 0.93 0.95</td>
<td>0.85 0.74 0.79</td>
<td>0.85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exact</td>
<td>0.96 0.93 0.75</td>
<td>0.72 0.74 0.85</td>
<td>0.77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Estimated precision and recall over seven languages.
seven languages and generated high quality semantics labels across the board. For predicate labels, the precision is over 95% and the recall is over 85% for all languages except for Hindi. For argument labels, when considering partially correct matches, the precision is at least 85% (above 90% for most languages) and the recall is between 66% to 83% for all the languages. These encouraging results obtained from a diverse set of languages implies the generalizability of our method. In addition, the inter-annotator agreement is very high for all the languages, indicating that the results obtained based on manual evaluation are very reliable.

In addition, we make a number of interesting observations:

**Dependency Parsing Accuracy** The precision for exact argument labels is significantly below partial matches, particularly for Hindi (↓35 pp) and Arabic (↓19 pp). Since argument boundaries are determined syntactically, such errors are caused by dependency parsing. The fact that Hindi and Arabic suffer the most from this issue is consistent with the poorer performance of their dependency parsers compared to other languages (Nivre et al., 2006; Green and Manning, 2010).

**Hindi as the Main Outlier** The results for Hindi are much worse than the results for other languages. Besides the poorer dependency parser performance, the size of the parallel corpus used could be a factor: Hindencorp is one to two orders of magnitude smaller than the other corpora. The quality of the parallel corpus could be a reason as well: Hindencorp was collected from various sources, while both UN and Europarl were extracted from governmental proceedings.

**Language-specific Errors** Certain errors occur more frequently in some languages than others. An example are deverbal nouns in Chinese (Xue, 2006) in formal passive constructions with support verb 受. Since we currently only consider verbs for pred-
ing language-specific errors. As an initial experiment, we added a simple heuristic to filter out French verbs that are commonly used for “existential there” constructions, as one type of common errors for French involves the syntactic expletive *il* (Danlos, 2005) in “existential there” constructions such as *il faut* (see Fig. 1 (TL sentence) for an example) wrongly labeled with with role information. As shown in Tab. 9, this simple customization results in a small increase in precision, suggesting that language-specific customization can be helpful.

**Quality of English SRL** As noted in Sec. 2.5, errors made by English SRL are often prorogated to the TL via projection. To assess the impact of English SRL quality, we used two different English SRL systems: CLEARNLP and MATE-SRL. As can be seen from Tab. 9, the impact of English SRL quality is substantial on argument labeling.

### 4.4 Multilingual PropBanks

To facilitate future research on multilingual SRL, we release the created PropBanks for all 7 languages to the research community to encourage further research. Tab. 7 gives an overview over the resources.

### 5 Related Work

**Annotation Projection in Parallel Corpora** to train monolingual tools for new languages was introduced in the context of learning a PoS tagger (Yarowsky et al., 2001). Similar in spirit to our approach of using filters to increase the precision of projected labels, recent work (Täckström et al., 2013) uses token and type constraints to guide learning in cross-lingual PoS tagging.

**Projection of Semantic Labels** was considered for FrameNet (Baker et al., 1998) in (Pado and Lapata, 2009). Recently, however, most work in the area focuses on PropBank, which has been identified as a more suitable annotation scheme for joint syntactic-semantics settings due to broader coverage (Merlo and van der Plas, 2009), and was shown to be usable for languages other than English (Monachesi et al., 2007).

Direct projection of PropBank annotations was considered in (Van der Plas et al., 2011). Our approach significantly outperforms theirs in terms of recall and $F_1$ for both predicates and arguments (Section 3). A approach was proposed in (Van der Plas et al., 2014) in which information is aggregated at the corpus level, resulting in a significantly better SRL corpus for French. However, this approach has several practical limitations: (1) it does not consider the problem of argument identification of SRL systems, treating arguments as already given; (2) it generates rules for the argument classification step preferably from manually annotated data; (3) it has been demonstrated for a single language (French), and was not applied to any other language. In contrast, our approach trains an SRL system for both predicate and argument labels, in a completely automatic fashion. Furthermore, we have applied our approach to generate PropBanks for 7 languages and conducted experiments that indicate a high $F_1$ measure for all languages (Section 4).

**Other Related Work** A number of approaches such as model transfer (Kozhevanikov and Titov, 2013) and role induction (Titov and Klementiev, 2012) exist for the argument classification step in the SRL pipeline. In contrast, our work addresses the full SRL pipeline and seeks to generate SRL resources for TLs with English PropBank labels.

### 6 Conclusion

We proposed a two-staged method to construct multilingual SRL resources using monolingual SRL and parallel data and showed that our method outperforms previous approaches in both precision and recall. More importantly, through comprehensive experiments over seven languages from three language families, we show that our proposed method works well across different languages without any language specific customization. Preliminary results from additional experiments indicate that better English SRL and language-specific customization can further improve the results, which we aim to investigate in future work. A qualitative comparison against existing or under-construction PropBanks for Chinese (Xue, 2008), Hindi (Vaidya et al., 2011) or Arabic (Zaghouani et al., 2010) may be interesting, both for comparison of resources and for defining language-specific customizations. In addition, we plan to expand our experiments both to more languages as well as NomBank (Meyers et al., 2004)-style noun labels.
References


