InVeRo-XL: Making Cross-Lingual Semantic Role Labeling Accessible with Intelligible Verbs and Roles

Simone Conia  
Sapienza University of Rome  
conia@di.uniroma1.it

Riccardo Orlando  
Babelscape, Italy  
orlando@babelscape.com

Fabrizio Brignone  
Babelscape, Italy  
brignone@babelscape.com

Francesco Cecconi  
Babelscape, Italy  
cecconi@babelscape.com

Roberto Navigli  
Sapienza University of Rome  
navigli@diag.uniroma1.it

Abstract

Notwithstanding the growing interest in cross-lingual techniques for Natural Language Processing, there has been a surprisingly small number of efforts aimed at the development of easy-to-use tools for cross-lingual Semantic Role Labeling. In this paper, we fill this gap and present InVeRo-XL, an off-the-shelf state-of-the-art system capable of annotating text with predicate sense and semantic role labels from 7 predicate-argument structure inventories in more than 40 languages. We hope that our system – with its easy-to-use RESTful API and Web interface – will become a valuable tool for the research community, encouraging the integration of sentence-level semantics into cross-lingual downstream tasks. InVeRo-XL is available online at http://nlp.uniroma1.it/invero.

1 Introduction

Informally, Semantic Role Labeling (SRL) is often defined as the task of automatically answering the question “Who did What, to Whom, Where, When, and How?” (Márquez et al., 2008). More precisely, SRL aims at recovering the predicate-argument structures within a sentence, providing an explicit overlay that uncovers the underlying semantics of text. For this reason, SRL is thought to be key in enabling Natural Language Understanding (Navigli, 2018). Today SRL is still an open problem, with several research papers being published each year at top-tier conferences, revealing novel insights and proposing better approaches. Over the years, thanks to this active development, SRL has been successfully exploited in a wide array of downstream tasks that span across different areas of Artificial Intelligence, from Natural Language Processing (NLP) with Information Retrieval (Christensen et al., 2010), Question Answering (He et al., 2015), Machine Translation (Marcheggiani et al., 2018) and Semantic Parsing (Banarescu et al., 2013), to Computer Vision with Visual Semantic Role Labeling (Gupta and Malik, 2015) and Situation Recognition (Yatskar et al., 2016).

Recently, the growing interest in cross-lingual NLP, supported by the increasingly wide availability of pretrained multilingual language models such as BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), has sparked renewed interest in multilingual and cross-lingual SRL. In just a few years, researchers have found ways to design fully-neural end-to-end systems for SRL (Cai et al., 2018), to take advantage of contextual word representations (Peters et al., 2018; Li et al., 2019), to achieve high performance on multiple languages (He et al., 2019a; Conia and Navigli, 2020), to generate sense and role labels with sequence-to-sequence models (Bilošhmi et al., 2021) and to perform SRL jointly across heterogeneous inventories (Conia et al., 2021).

Since SRL is a task that involves complex linguistic theories, inventories and techniques, there have been efforts to develop easy-to-use tools that offer automatic predicate sense and semantic role annotations to users interested in the integration of sentence-level semantics into downstream tasks. Some notable examples include SENNA1 (Collobert et al., 2011), which uses an ensemble of feature-based classifiers (Koomen et al., 2005), AllenNLP’s SRL demo2, which provides a reimplementation of a BERT-based model (Shi and Lin, 2019), and InVeRo (Conia et al., 2020), which offers annotations according to two different linguistic inventories, PropBank (Palmer et al., 2005) and

1https://ronan.collobert.com/senna
2https://demo.allennlp.org/semantic-role-labeling
VerbAtlas (Di Fabio et al., 2019). However, one important drawback of the above-mentioned tools is that they are able to perform SRL only in English, which hinders the exploitation of their annotations in multilingual and cross-lingual NLP.

In order to fill this gap, we build upon InVeRo and propose its next major release, InVeRo-XL, with the objective of making SRL accessible in multiple languages. We rebuild InVeRo-XL from the ground up to offer:

- The first end-to-end system to tackle the whole SRL pipeline in over 40 languages;
- The first off-the-shelf system to provide SRL annotations for 7 linguistic inventories;
- A RESTful API service that can be queried either online, so as not to install any software, or offline, to maximize throughput;
- A Web interface that provides a visualization of the system output which can be useful for teaching purposes, comparing linguistic theories, and prototyping new ideas.

We believe that InVeRo-XL can provide a stepping stone for the integration of explicit sentence-level semantics into cross-lingual tasks, attracting new researchers to the field of SRL and its applications.3

1.1 What’s New in InVeRo-XL

As previously mentioned, InVeRo-XL is the successor of InVeRo. Although its main new feature is the ability to provide predicate sense and semantic role annotations in over 40 languages with 7 different inventories, InVeRo-XL has been overhauled to also improve several other important aspects. In particular:

- **Preprocessing:** while its predecessor used a very limited set of rules to preprocess English text, InVeRo-XL features a multilingual preprocessing module based on spaCy and Stanza;
- **SRL model:** the English-only model has been replaced by a cross-lingual model that is able to perform not only span-based SRL, but also dependency-based SRL;
- **API:** the service is now able to handle batched requests and documents of arbitrary length;
- **Offline usage:** InVeRo-XL is now available for download, free for research purposes, allowing users to host their own instance locally.

2 System Overview

In this Section, we provide an overview of the main components of InVeRo-XL and how they interact, describing in detail the preprocessing module (Section 2.1) and the SRL model (Section 2.2).

2.1 Preprocessing

The previous version of InVeRo-XL preprocessed an English sentence using a very limited and simple set of rules. In order to correctly support more languages, InVeRo-XL now relies on both spaCy (Honnibal et al., 2020) and Stanza (Qi et al., 2020) to deal transparently with document splitting and tokenization. An automatic language detector based on fastText4 (Joulin et al., 2017) is used to dynamically choose between the two preprocessing tools, depending on the language detected: spaCy is faster for high-resource languages, e.g. English, but also less reliable on lower-resource languages, e.g. Catalan, for which our system falls back to Stanza.

2.2 Model Architecture

In line with its predecessor, InVeRo-XL encapsulates an SRL model that falls within the broad category of end-to-end systems, tackling the whole SRL pipeline – predicate identification, predicate sense disambiguation, argument identification and argument classification – in a single forward pass. However, the design of the SRL model itself has been completely revamped and now follows the architecture recently proposed by Conia et al. (2021), which is capable of performing cross-lingual SRL with heterogeneous linguistic inventories. In the following, we describe the main components of the SRL model architecture provided by InVeRo-XL.

Multilingual word encoder. The first component of our model is a multilingual word encoder that takes advantage of a pretrained language model, XLM-RoBERTa (Conneau et al., 2020), to provide rich contextualized word representations.

3InVeRo-XL can be downloaded upon request at http://nlp.uniroma1.it/resources. InVeRo-XL is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International.

4https://fasttext.cc
More formally, for each word \( w_i \) in an input sentence \( w = \langle w_1, w_2, \ldots, w_n \rangle \) of length \( n \), it computes an encoding \( e_i = \text{Swish}(W^e h_i + b^w) \) as a non-linear projection of the concatenation \( h_i \) of the corresponding hidden states of the four topmost layers of the language model.

**Universal word encoder.** The resulting sequence of multilingual word encodings \( E = \langle e_1, e_2, \ldots, e_n \rangle \) is then given to a “universal” word encoder that computes a sequence of task-specific timestep encodings \( T = \langle t_1, t_2, \ldots, t_n \rangle \) as follows:

\[
t^j_i = \begin{cases} 
e_i & \text{if } j = 0 \\
 t^{j-1}_i \oplus \text{BiLSTM}_j^i(t^{j-1}) & \text{otherwise}
\end{cases}
\]

\[
T = \langle t^K_1, t^K_2, \ldots, t^K_n \rangle
\]

where \( \text{BiLSTM}_j^i(\cdot) \) is the \( i \)-th timestep of the \( j \)-th BiLSTM layer and \( K \) is the total number of BiLSTM layers. The purpose of this encoder is to create representations that are shared across languages and inventories and are, therefore, “universal”.

**Universal predicate-argument encoder.** Similarly to the encoder above, the objective of the universal predicate-argument encoder is to build predicate-specific argument representations that lie in a vector space shared across languages and inventories. Assuming that \( w_p \) is a predicate in the input sentence \( w \), this encoder builds a sequence \( A \) of predicate-specific argument encodings as follows:

\[
a^j_i = \begin{cases} 
t_p \oplus t_i & \text{if } j = 0 \\
 a^{j-1}_i \oplus \text{BiLSTM}_j^i(a^{j-1}_i) & \text{otherwise}
\end{cases}
\]

\[
A = \langle a^K_1, a^K_2, \ldots, a^K_n \rangle
\]

where \( t_i \) is the \( i \)-th timestep encoding from the universal sentence encoder, \( a^j_i \) is the argument encoding for \( w_i \) with respect to \( w_p \) produced after the \( j \)-th BiLSTM layer, and \( K' \) is the total number of BiLSTM layers.

**Inventory-specific decoders.** Finally, the universal encodings are given to a set of classifiers in order to obtain the desired output labels. More specifically, for each inventory, we need three types of output: i) whether a word \( w_i \) is a predicate \( w_p \); ii) the most appropriate sense \( s \) for a predicate \( w_p \); iii) which semantic role \( r \), possibly the null role, exists between a word \( w_i \) and a predicate \( w_p \). More formally, our model features three classifiers for each inventory \( I \) as follows:

\[
\sigma^p(i | w_i, I) = W^{p|I}\text{Swish}(W^p t_i + b^p) + b^{p|I}
\]

\[
\sigma^s(i | w_p, I) = W^{s|I}\text{Swish}(W^s t_p + b^s) + b^{s|I}
\]

\[
\sigma^r(i | w_p, I) = W^{r|I}\text{Swish}(W^r a_i + b^r) + b^{r|I}
\]

where each \( \sigma(\cdot) \) provides a score distribution over the possible output classes, i.e. two (true or false) for predicate identification, the number of senses of an inventory for predicate sense disambiguation, and the number of semantic roles (including the null role) of an inventory for argument labeling.

**Miscellanea.** While we follow the architecture proposed by Conia et al. (2021), the SRL model of InVeRo-XL also comes with a small but significant number of enhancements. One such enhancement is that, while Conia et al. (2021) propose a model for dependency-based SRL, our model is also able to perform span-based SRL by treating spans as sequences of BIO tags. In order to correctly decode valid spans at inference time, InVeRo-XL makes use of a Viterbi decoder. Other improvements include training the model with the RAAdam optimizer (Liu et al., 2020), ensuring that each training batch features a balanced number of instances for each language in the training set, and searching randomly for better hyperparameter values.

### 2.3 Evaluation

**Datasets.** We report the performance of InVeRo-XL on two gold standard benchmarks for SRL: CoNLL-2009 (Hajic et al., 2009) for dependency-based SRL and CoNLL-2012 (Pradhan et al., 2012) for span-based SRL. To the best of our knowledge, CoNLL-2009 is the largest benchmark for multilingual SRL as it comprises six languages, namely, Catalan, Chinese, Czech, English, German and Spanish.\(^3\) The main challenge of this benchmark is that each language was annotated with a different predicate-argument structure inventory, e.g. the English PropBank (Palmer et al., 2005) for English, AnCora (Taulé et al., 2008) for Spanish/Catalan and PDT-Vallex (Hajic et al., 2003) for Czech. While CoNLL-2009 is an ideal test bed for evaluating the multilingual capabilities of an SRL system, dependency-based annotations may look unfamiliar to end users who are not used to the

\(^3\)Japanese is not available anymore from LDC due to licensing issues.
Table 1: Comparison between InVeRo-XL and other recent systems for SRL. 

<table>
<thead>
<tr>
<th></th>
<th>Catalan</th>
<th>Czech</th>
<th>German</th>
<th>English</th>
<th>Spanish</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllenNLP’s SRL demo</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>86.5</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>InVeRo</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>86.2</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>InVeRo-XL\textsubscript{span-based}</td>
<td><strong>83.3</strong></td>
<td><strong>85.9</strong></td>
<td><strong>87.0</strong></td>
<td><strong>86.8</strong></td>
<td><strong>81.8</strong></td>
<td><strong>84.9</strong></td>
</tr>
<tr>
<td>Marcheggiani et al. (2017)</td>
<td>–</td>
<td>86.0</td>
<td>–</td>
<td>87.7</td>
<td>80.3</td>
<td>81.2</td>
</tr>
<tr>
<td>Chen et al. (2019)</td>
<td>81.7</td>
<td>88.1</td>
<td>76.4</td>
<td>91.1</td>
<td>81.3</td>
<td>81.7</td>
</tr>
<tr>
<td>Cai and Lapata (2019b)</td>
<td>–</td>
<td>–</td>
<td>82.7</td>
<td>90.0</td>
<td>81.8</td>
<td>83.6</td>
</tr>
<tr>
<td>Cai and Lapata (2019a)</td>
<td>–</td>
<td>–</td>
<td>83.8</td>
<td>91.2</td>
<td>82.9</td>
<td>85.0</td>
</tr>
<tr>
<td>Lyu et al. (2019)</td>
<td>80.9</td>
<td>87.5</td>
<td>75.8</td>
<td>90.1</td>
<td>80.5</td>
<td>83.3</td>
</tr>
<tr>
<td>He et al. (2019b)</td>
<td>86.0</td>
<td>89.7</td>
<td>81.1</td>
<td>90.9</td>
<td>85.2</td>
<td>86.9</td>
</tr>
<tr>
<td>Conia and Navigli (2020)</td>
<td>88.3</td>
<td><strong>92.1</strong></td>
<td>89.1</td>
<td><strong>92.4</strong></td>
<td>86.9</td>
<td><strong>89.1</strong></td>
</tr>
<tr>
<td>Conia et al. (2021)</td>
<td>88.0</td>
<td>91.5</td>
<td>88.0</td>
<td>91.8</td>
<td>86.3</td>
<td>87.7</td>
</tr>
<tr>
<td>InVeRo-XL\textsubscript{dependency-based}</td>
<td><strong>88.7</strong></td>
<td><strong>92.1</strong></td>
<td><strong>89.9</strong></td>
<td><strong>92.1</strong></td>
<td><strong>87.2</strong></td>
<td><strong>89.1</strong></td>
</tr>
</tbody>
</table>

Table 1: Comparison between InVeRo-XL and other recent systems for SRL. 

**Top:** $F_1$ scores on argument labeling with pre-identified predicates using the official CoNLL-2005 scoring script on the CoNLL-2012 English test set for span-based SRL and the CoNLL-2009 test sets converted from dependency-based to span-based as described in Section 2.3. 

**Bottom:** $F_1$ scores on argument labeling and sense disambiguation with pre-identified predicates using the official CoNLL-2009 scoring script on the test sets of the CoNLL-2009 shared task for dependency-based multilingual SRL.

The InVeRo-XL API

In order to facilitate the integration of predicate-argument structure information into downstream tasks, InVeRo-XL exposes its fully self-contained end-to-end multilingual SRL pipeline through an easy-to-use RESTful API. In the following, we provide an overview of the main functionalities of the InVeRo-XL API, from its Resource API (Section 3.1) to its Model API (Section 3.2) and how to host InVeRo-XL locally on a user’s own hardware (Section 3.3). We refer users to the online documentation for the complete list of supported languages.
and inventories, together with other details.\footnote{http://nlp.uniroma1.it/invero/api-documentation}

### 3.1 Resource API

The Resource API is a simple way for obtaining semantic information about predicates using the intelligible verb senses and semantic roles defined by VerbAtlas, a large-scale predicate-argument structure inventory which clusters WordNet synsets (Miller, 1992) that share similar semantic behavior. The Resource API of InVeRo-XL builds upon the functionalities provided by its predecessor with the key difference that it now supports multiple languages thanks to BabelNet 5.0\footnote{https://babelnet.org} (Navigli and Ponzetto, 2012; Navigli et al., 2021), a multilingual encyclopedic dictionary that provides unified access to several knowledge bases including WordNet.

More specifically, the Resource API defines two endpoints:

- `/api/verbatlas/predicate`: given a predicate $p$, this endpoint retrieves the set of VerbAtlas frames which include at least one sense of $p$.
- `/api/verbatlas/frame`: given a VerbAtlas frame $f$, this endpoint retrieves its predicate-argument structure, i.e., the semantic roles, and the WordNet/BabelNet synsets that belong to $f$.

### 3.2 Model API

The Model API of InVeRo-XL has been updated to not only take advantage of the new multilingual SRL system but also to provide quality-of-life improvements. The Model API now accepts requests in over 40 languages and returns semantic annotations according to 7 linguistic inventories. On top of this, the Model API is now able to process documents of arbitrary length and to handle batches of documents in a single request.

More specifically, the Model API exposes an endpoint named `/api/model`. This endpoint accepts POST requests with a JSON body containing a list of input objects, one for each document the user wishes to annotate. Each input object shall specify the following fields:

- `text`: a mandatory field that contains the text of the document.
- `lang`: an optional field that indicates the language of the document. If omitted, InVeRo-XL will use an automatic language detector (see Section 2.1).

Each request to the Model API returns a JSON response containing a list of output objects, one for each input document, containing the automatic annotations according to each of the 7 linguistic inventories, as shown in Figure 1.
Figure 2: The home page of the Web interface of InVeRo-XL. Users can search for predicate information (e.g., the VerbAtlas frames a verb belongs to) and tag sentences in multiple languages with different linguistic inventories (see Figure 3).

3.3 Offline Usage

One of the most requested features that is currently missing from InVeRo is the possibility of running an offline instance of the service so as to annotate large quantities of text in a shorter time, independently of the latency of the network and the volume of requests being processed by our Web server. To address this issue, InVeRo-XL is also distributed as a Docker image that can be deployed locally on a user’s own hardware. While network latency is often a bottleneck for processing a request, an offline instance of InVeRo-XL does not suffer from such a constraint and can therefore benefit greatly from running on better hardware, e.g., on GPU. We distribute InVeRo-XL in two configurations:

- `invero-xl-span` is the configuration that performs multilingual span-based SRL and is the one used by InVeRo-XL’s Web server;

- `invero-xl-dependency` is an alternative configuration built to perform multilingual dependency-based SRL.

Running a local instance of InVeRo-XL is also simple. First, users are required to perform a one-time setup to load one of the available images:

```bash
#!/bin/bash
docker load -i invero-xl-span_2.0.0.tar
```

After that, InVeRo-XL can be started with:

```bash
#!/bin/bash
PORT=12345
LANGUAGES="EN IT FR ZH"
docker run \
  --name invero-xl-span-en \
  -p $PORT:80 \
  -e LANGUAGES=$LANGUAGES \
invero-xl-span:2.0.0
```

Once started, users can forward their requests locally. We refer the reader to the online documentation for further details.10

4 Web Interface

Similarly to its predecessor, InVeRo-XL includes a public-facing Web interface (Figures 2 and 3) that provides a visual environment for both the Resource API and the Model API, allowing users to explore the main functionalities while also providing an intuitive overview of how an SRL system annotates a sentence or a short document. Most importantly, the Web interface of InVeRo-XL has been updated to reflect the changes in the Model API and the underlying SRL model; now users can annotate text in 12 languages11 and visualize predicate senses and semantic roles in 7 linguistic inventories on the fly, without having to write code.

Figure 3 shows an example sentence in Italian with its corresponding predicate senses and semantic roles.
tic roles as provided by InVeRo-XL. Thanks to a dropdown menu, users can immediately switch from the labels of one inventory to those of another, independently of the input language, without reloading the Web page. We argue that this Web interface should help teachers explain SRL to their students, allow linguists to compare linguistic inventories on particular case studies, attract new researchers to the field, and inspire others to exploit SRL in downstream tasks or even real-world scenarios.

5 Conclusion and Future Work

Over the years, the research community has greatly advanced the field of SRL, proposing ever more complex approaches to tackle the task more effectively. However, despite the growing interest in cross-lingual NLP, there have been very few efforts to develop automatic tools to perform SRL in multiple languages. Our objective with InVeRo-XL is to fill this gap and equip researchers with an easy-to-use, high-performing system capable of providing predicate sense and semantic role annotations in over 40 languages with 7 linguistic inventories. Users can take advantage of our state-of-the-art system for cross-lingual SRL through a RESTful API that relieves them from the need to reimplement complex neural models and/or to build an efficient preprocessing/postprocessing pipeline.

Although InVeRo-XL is a major step forward compared to its predecessor, we intend to further improve our system by adopting future and more advanced SRL model architectures and by including new training datasets, such as UniteD-SRL (Tripodi et al., 2021). We strongly believe that InVeRo-XL will facilitate the integration of SRL into downstream cross-lingual tasks, hopefully aiding further advancements in cross-lingual Natural Language Understanding.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 and the European Language Grid project No. 825627 (Universal Semantic Annotator, USeA) under the European Union’s Horizon 2020 research and innovation programme.
References


Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven Semantic Role Labeling:


