# A flexible and easy-to-use semantic role labeling framework for different languages

**Quynh Ngoc Thi Do  Artuur Leeuwenberg  Geert Heyman  Marie-Francine Moens**
Department of Computer Science, KU Leuven, Belgium
{quynhngocthi.do,tuur.leeuwenberg,geert.heyman,sien.moens}@cs.kuleuven.be

## Abstract

This paper presents *DAMESRL*[1], a flexible and open source framework for deep semantic role labeling (SRL). DAMESRL aims to facilitate easy exploration of model structures for multiple languages with different characteristics. It provides flexibility in its model construction in terms of word representation, sequence representation, output modeling, and inference styles and comes with clear output visualization. Additionally, it handles various input and output formats and comes with clear output visualization. The framework is available under the Apache 2.0 license.

## 1 Introduction

During the first decade of the $21^{st}$ century, mapping from the syntactic analysis of a sentence to its semantic representation has received a central interest in the natural language processing (NLP) community. Semantic role labeling, which is a sentence-level semantic task aimed at identifying "Who did What to Whom, and How, When and Where?" (Palmer et al., 2010), has strengthened this focus. Recently, several neural mechanisms have been used to train end-to-end SRL models that do not require task-specific feature engineering as the traditional SRL models do. Zhou and Xu (2015) introduced the first deep end-to-end model for SRL using a stacked Bi-LSTM network with a conditional random field (CRF) as the top layer. He et al. (2017) simplified their architecture using a highway Bi-LSTM network. More recently, Tan et al. (2018) replaced the common recurrent architecture with a self-attention network, directly capturing relationships between tokens regardless of their distance, resulting in better results and faster training. The work in deep end-to-end SRL has focused heavily on applying deep learning advances without considering the multilingual aspect. However, language-specific characteristics and the available amount of training data highly influence the optimal model structure.

DAMESRL facilitates exploration and fair evaluation of new SRL models for different languages by providing flexible neural model construction on different modeling levels, the handling of various input and output formats, and clear output visualization. Beyond the existing state-of-the-art models (Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018), we exploit character-level modeling, beneficial when considering multiple languages. To demonstrate the merits of easy cross-lingual exploration and evaluation of model structures for SRL provided by DAMESRL, we report performance of several distinct models integrated into our framework for English, German and Arabic, as they have very different linguistic characteristics.

## 2 Task Definition

Formally, the goal of end-to-end SRL is to predict a sequence $(l_1, l_2, \ldots, l_n)$ of semantic labels given a sentence $(w_1, w_2, \ldots, w_n)$, and its predicate $w_p$ as input. Each $l_i$, which belongs to a discrete set of PropBank BIO tags, is the semantic tag corresponding to the word $w_i$ in the semantic frame evoked

[1]The source code can be found at: https://liir.cs.kuleuven.be/software_pages/damesrl.php.





*Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 161–165
Santa Fe, New Mexico, USA, August 20-26, 2018.

by $w_p$. Here, words outside argument spans have the tag $\mathbf{O}$, and words at the beginning and inside of argument spans with role $r$ have the tags $\mathbf{B_r}$ and $\mathbf{I_r}$, respectively. For example, the sentence "the cat chases the dog ." should be annotated as "the$_{\mathbf{B-A0}}$ cat$_{\mathbf{I-A0}}$ chases$_{\mathbf{B-V}}$ the$_{\mathbf{B-A1}}$ dog$_{\mathbf{I-A1}}$ .$_{\mathbf{O}}$".

## 3 System Architecture

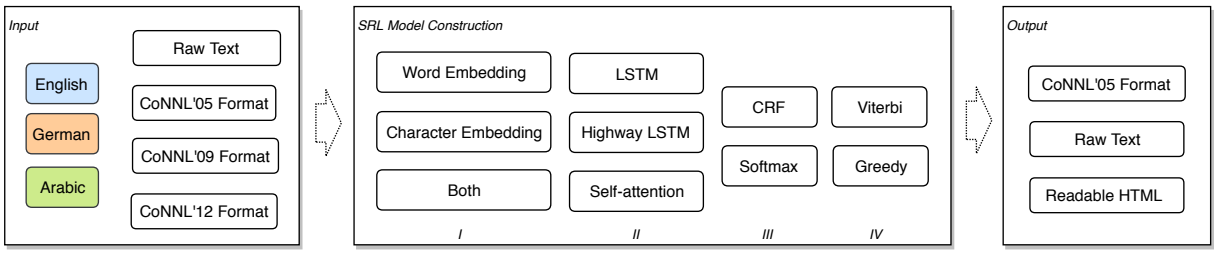


Figure 1: Schematic overview of the DAMESRL architecture from input to output.

DAMESRL's architecture (see Fig. 1) facilitates the construction of models that prioritize certain language-dependent linguistic properties, such as the importance of word order and inflection, or that adapt to the amount of available training data. The framework, implemented in Python 3.5 using TensorFlow, can be used to train new models, or make predictions with the provided pre-trained models.

### 3.1 Input and Output

The input/output format of DAMESRL is a shortened version of the CoNLL'05 format, which only contains the Words, Targets and (possibly) Props columns[2]. DAMESRL also provides an HTML format that can be directly visualized in the web browser (as in Fig. 2).

### 3.2 Model Construction Modules

As can be seen in Fig. 1, the framework divides model construction in four phases: (I) word representation, (II) sentence representation, (III) output modeling, and (IV) inference.

**Phase I:** The word representation of a word $w_i$ consist of three optional concatenated components: a word-embedding, a Boolean indicating if $w_i$ is the predicate of the semantic frame ($w_p$), and a character representation. DAMESRL provides a Bi-LSTM network to learn character-level word representations helping for languages where important SRL cues are given through inflections, such as case markings in German and Arabic. Despite the foreseen importance, character-level embeddings have not been used in previous work (Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018).

**Phase II:** As core sequence representation component, users can choose between a self-attention encoding (Tan et al., 2018), a regular Bi-LSTM (Hochreiter and Schmidhuber, 1997) or a highway Bi-LSTM (Zhang et al., 2016; He et al., 2017).

**Phase III:** To compute model probabilities, users can choose a regular softmax, or a linear chain CRF as proposed by (Zhou and Xu, 2015), which can be useful for languages where word order is an important SRL cue, such as English, or when less training data is available (shown in Section 4).

[2]http://www.lsi.upc.edu/šrlconll/conll05st-release/README

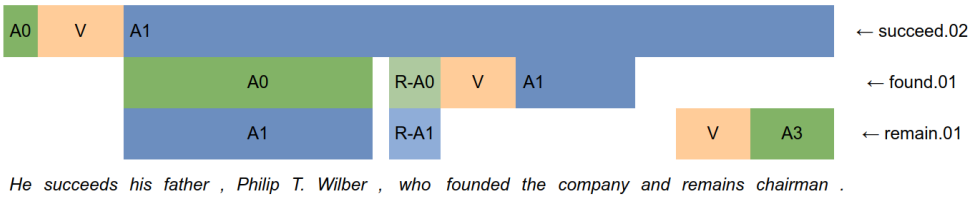


Figure 2: Screen-shot of the HTML Output

**Phase IV:** The inference phase provides two options for label inference from the computed model probabilities including greedy prediction and Viterbi decoding.

## 4 Experiments

### 4.1 Settings

To evaluate our framework, and show the benefits of choosing certain model components, we construct five models: HLstm, Char, CRFm, Att, and CharAtt, whose configurations are shown in Tab. 1. The

Table 1: Configurations of experimental models.

| | HLstm | Char | CRFm | Att | CharAtt |
|---|---|---|---|---|---|
| Word Emb. | ✓ | | ✓ | ✓ | |
| Word + Character Emb. | | ✓ | | | ✓ |
| Highway LSTM | ✓ | ✓ | ✓ | | |
| Self-Attention | | | | ✓ | ✓ |
| Softmax | ✓ | ✓ | | ✓ | ✓ |
| CRF | | | ✓ | | |

Table 2: Training data.

| | English | German | Arabic |
|---|---|---|---|
| Source | CoNLL'05 | CoNLL'09 | CoNLL'12 |
| # Sentences | 39832 | 36020 | 7422 |
| Vocab. size | 35094 | 67495 | 45683 |
| # Predicates | 90750 | 17400 | 20001 |

selected models are evaluated in three languages: English, German and Arabic (see Tab. 2) using the standard CoNLL'05 metrics. Information about the used SRL data is shown in Tab. 2. We initialize the weights of all sub-layers as random orthogonal matrices. The learning rate is fixed in the first $N_1$ training epochs, and halved after every next $N_2$ epochs. Detailed settings and the word embeddings used to initialize the word representation layer used per language are found in Tab. 3.

Table 3: Experimental settings.

| Setting | Model | Value |
|---|---|---|
| Optimizer | All | AdaDelta, $\epsilon = 1e{-}06$ |
| Learning rate | All | 1.0 |
| Dropout probability | All | 0.1 |
| Label smoothing value | All | 0.1 |
| Word-emb size | All | 100 |
| Word-emb type | All | GloVe |
| Batch size | All | 80 predicates |
| Early stopping patience | All | 100 |
| $N_1$ | HLstm, Char, Att, CharAtt | 400 |
| $N_2$ | HLstm, Char, Att, CharAtt | 100 |
| $N_1$ | CRFm | 100 |
| $N_2$ | CRFm | 30 |
| # Max. training epochs | Att, CharAtt | 800 |
| # Hidden layers | Att, CharAtt | 10 |
| # Max. training epochs | HLstm, Char, CRFm | 500 |
| # Hidden layers | HLstm, Char, CRFm | 8 |
| Hidden layer size | HLstm, Char, CRFm | 300 |
| Character-emb. size | Char, CharAtt | 100 |
| Position Encoding | Att, CharAtt | Timing |
| Word-emb. data | English | Wikipedia+Gigaword[3] |
| Word-emb. data | German | Wikipedia |
| Word-emb. data | Arabic | None |

Table 4: Training (Tr.) and prediction (Pr.) times (greedy) for English.

| | Tr. time / epoch | Pr. time / predicate |
|---|---|---|
| HLstm | 10 mins | 8.5 ms |
| Char | 12 mins | 15.5 ms |
| CRFm | 8 mins | 11.4 ms |
| Att | 2 mins | 3.4 ms |
| CharAtt | 5 mins | 4.2 ms |

[3]From: https://nlp.stanford.edu/projects/glove/

Table 5: Results on CoNLL'12 Arabic and CoNLL'09 German data: precision (P), recall (R), and F1.[4]

| Model | Arabic | | | | | | German | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Development | | | Evaluation | | | Development | | | Out-Of-Domain | | | Evaluation | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| HLstm | 46.2 | 45.2 | 45.7 | 47.4 | 45.3 | 46.3 | 67.9 | 66.4 | 67.1 | 55.6 | 57.2 | 56.4 | 68.2 | 67.1 | 67.6 |
| Char | 51.2 | 50.2 | 50.7 | 47.5 | 46.0 | 46.7 | 69.1 | 66.5 | 67.8 | 54.0 | 55.2 | 54.6 | 68.2 | 67.0 | 67.6 |
| CRFm | 50.8 | 47.7 | 49.2 | 51.9 | 48.0 | 49.9 | 68.7 | 66.0 | 67.3 | 55.3 | 53.8 | 54.6 | 65.8 | 64.4 | 65.1 |
| Att | 50.4 | 48.0 | 49.2 | 50.0 | 46.7 | 48.3 | 71.6 | 70.8 | 71.2 | 54.7 | 56.8 | 55.7 | 71.9 | 71.5 | 71.7 |
| CharAtt | 56.9 | 56.0 | **56.5** | 56.0 | 54.5 | **55.2** | 74.8 | 73.8 | **74.3** | 57.2 | 57.3 | **57.3** | 73.4 | 73.6 | **73.5** |

Table 6: Results on CoNLL'05 English data: precision (P), recall (R), and F1. We compare our results with other state-of-the-art deep *single* models.

| Model | Development | | | Out-Of-Domain | | | Evaluation | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Lstm + CRF (Zhou and Xu, 2015) | 79.7 | 79.4 | 79.6 | 70.7 | 68.2 | 69.4 | 82.9 | 82.8 | 82.8 |
| HLstm (He et al., 2017) | 81.6 | 81.6 | 81.6 | 72.9 | 71.4 | 72.1 | 83.1 | 83.0 | 83.1 |
| Att (Tan et al., 2018) | 82.6 | 83.6 | 83.1 | 73.5 | 74.6 | **74.1** | 84.5 | 85.2 | 84.8 |
| HLstm-ours | 82.2 | 81.9 | 82.0 | 72.6 | 71.2 | 71.9 | 83.4 | 82.8 | 83.1 |
| Char | 82.3 | 82.1 | 82.2 | 73.3 | 71.7 | 72.5 | 83.8 | 82.9 | 83.4 |
| CRFm | 81.9 | 81.5 | 81.7 | 72.0 | 69.6 | 70.9 | 84.0 | 83.1 | 83.5 |
| Att-ours | 83.0 | 83.4 | 83.2 | 74.5 | 72.9 | *73.7* | 84.8 | 84.7 | 84.8 |
| CharAtt | 83.6 | 83.5 | **83.5** | 73.5 | 72.6 | 73.0 | 85.0 | 84.8 | **84.9** |

Table 7: F1 scores on CoNLL'05 English data, CoNLL'09 German data and CoNLL'12 Arabic data using 2000 random training sentences: Dev (Development), Eval (Evaluation), and Ood (Out of Domain).

| Model | English | | | German | | | Arabic | |
|---|---|---|---|---|---|---|---|---|
| | Dev | Eval | Ood | Dev | Eval | Ood | Dev | Eval |
| HLstm-ours | 62.8 | 54.3 | 64.9 | 42.34 | 45.58 | 40.44 | 35.12 | 34.42 |
| Char | 64.8 | 55.2 | 65.8 | **43.99** | **47.64** | **42.39** | 36.52 | 37.01 |
| CRFm | **65.8** | **57.5** | **67.0** | 43.42 | 44.06 | 40.73 | **38.91** | **38.36** |
| Att-ours | 57.4 | 51.7 | 59.6 | 32.48 | 37.13 | 31.45 | 23.38 | 23.32 |
| CharAtt | 58.2 | 52.4 | 60.7 | 33.35 | 38.49 | 31.91 | 35.10 | 34.70 |

### 4.2 Results and Discussion

In Tab. 5-6, we compare the five models on English, German and Arabic. The proposed CharAtt outperforms all other models in almost all cases except the English out-of-domain setting. As can be seen in Tab. 6, our implementation achieves competitive performance to other state-of-the-art systems for English. To the best of our knowledge, we report the first SRL results (in CoNLL'05 metrics) for German and Arabic without using linguistic features.

In general, we find that using character embeddings improves the performance of HLstm and Att, although at a cost of increased processing time. Interestingly, using character embeddings is particularly effective for the Att model. One explanation could be that character embeddings are important for learning good attention masks as they encode information about the syntax of words and the sentence, e.g., it facilitates the system in learning that the number (singular/plural) of a subject and its verb should match.

Among the three languages, the performance gain by character-level representations is larger for German and Arabic than for English. This can be explained by the much larger vocabularies for German and Arabic combined with the smaller training datasets (#sentences, and #predicates) for these languages. Moreover, many grammatical cases, which are very strong predictors for semantic roles, are explicitly

[4]Note that the CoNLL'09 data is automatically converted to CoNLL'05 format using the script by Björkelund et al. (2009).

marked through use of inflection in German and Arabic.

To evaluate the influence of the training size on model performance, we train the models on a random sample of 2000 sentences for each language (see Tab. 7). Intriguingly, the attention models perform worst in this setting, indicating their need of large datasets. A reason for this could be that the attention models consider the sequential dependency between hidden states to a lesser degree than recurrent models do. In contrast, CRFm achieves the best results for English and Arabic, and the second best result for German. In fact, CRFm exploits not only the input sequence – using the LSTM – but also the sequential output dependencies, to compute output probabilities. We can see that this is very beneficial when less training data is available, especially when word order is a strong cue for SRL, which applies well for a strict word order language like English. For such cases the output dependencies can be learned even from less training data, which results in the CRFm model to excel. As can be seen in Tab. 7, when comparing Char with HLstm-ours and CharAtt with Att-ours, the benefit of using character embeddings is demonstrated on small datasets as well.

## 5 Conclusions

We introduced an open source SRL framework, DAMESRL, which provides flexible model construction, using state-of-the-art model components, handles various input and output formats, and which comes with clear output visualization. Using our framework, we slightly improve the state-of-the-art results of single end-to-end deep systems on the English CoNLL'05, and report the first experimental end-to-end deep SRL results for German[5] and Arabic[5]. We have shown that the flexible model construction provided by the framework is crucial for exploring good model structures when considering different languages with different characteristics, especially when training data is limited. DAMESRL is made available under the Apache 2.0 license.

## References


Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of CoNLL: Shared Task*, CoNLL '09. ACL.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of ACL*, volume 1. ACL.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8).

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic role labeling*, volume 3. Morgan & Claypool Publishers.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *Proceedings of the ICLR Workshop*.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention.

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. 2016. Highway long short-term memory RNNs for distant speech recognition. In *Proceedings of ICASSP*. IEEE.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of ACL-IJCNLP*, volume 1. ACL.


[5]To the best of our knowledge, no results have been reported using CoNLL'05 metrics on these data. Pereyra et al. (2017) only report precision for argument classification for Arabic instead of using the standard CoNLL'05 metrics.