# CUNIT: A Semantic Role Labeling System for Modern Standard Arabic

**Mona Diab**
Columbia University
mdiab@cs.columbia.edu

**Alessandro Moschitti**
University of Trento, DIT
moschitti@dit.unitn.it

**Daniele Pighin**
FBK-irst; University of Trento, DIT
pighin@itc.it

## Abstract

In this paper, we present a system for Arabic semantic role labeling (SRL) based on SVMs and standard features. The system is evaluated on the released SEMEVAL 2007 development and test data. The results show an $F_{\beta=1}$ score of 94.06 on argument boundary detection and an overall $F_{\beta=1}$ score of 81.43 on the complete semantic role labeling task using gold parse trees.

## 1 Introduction

There is a widely held belief in the computational linguistics field that identifying and defining the roles of predicate arguments, semantic role labeling (SRL), in a sentence has a lot of potential for and is a significant step towards the improvement of important applications such as document retrieval, machine translation, question answering and information extraction. However, effective ways for seeing this belief come to fruition require a lot more research investment.

Since most of the available data resources are for the English language, most of the reported SRL systems to date only deal with English. Nevertheless, we do see some headway for other languages, such as German and Chinese (Erk and Pado, 2006; Sun and Jurafsky, 2004; Xue and Palmer, 2005). The systems for non-English languages follow the successful models devised for English, e.g. (Gildea and Jurafsky, 2002; Xue and Palmer, 2004; Pradhan et al., 2003). However, no SRL system exists for Arabic.

In this paper, we present a system for semantic role labeling for modern standard Arabic. To our knowledge, it is the first SRL system for a semitic language in the literature. It is based on a supervised model that uses support vector machines (SVM) technology for argument boundary detection and argument classification. It is trained and tested using the pilot Arabic PropBank data released as part of the `SEMEVAL 2007` data. Given the lack of a reliable deep syntactic parser, in this research we use gold trees.

The system yields an F-score of 94.06 on the sub task of argument boundary detection and an F-score of 81.43 on the complete task, i.e. boundary plus classification.

## 2 SRL system for Arabic

The design of an optimal model for an Arabic SRL systems should take into account specific linguistic aspects of the language. However, a remarkable amount of research has already been done in SRL and we can capitalize from it to design a basic and effective SRL system. The idea is to use the technology developed for English and verify if it is suitable for Arabic.

Our adopted SRL models use Support Vector Machines (SVM) to implement a two steps classification approach, i.e. boundary detection and argument classification. Such models have already been investigated in (Pradhan et al., 2003; Moschitti et al., 2005) and their description is hereafter reported.

### 2.1 Predicate Argument Extraction

The extraction of predicative structures is carried out at the sentence level. Given a predicate within a natural language sentence, its arguments have to be properly labeled. This problem is usually divided in two subtasks: (a) the detection of the boundaries, i.e. the word spans of the arguments, and (b) the classification of their type, e.g. *Arg0* and *ArgM* in

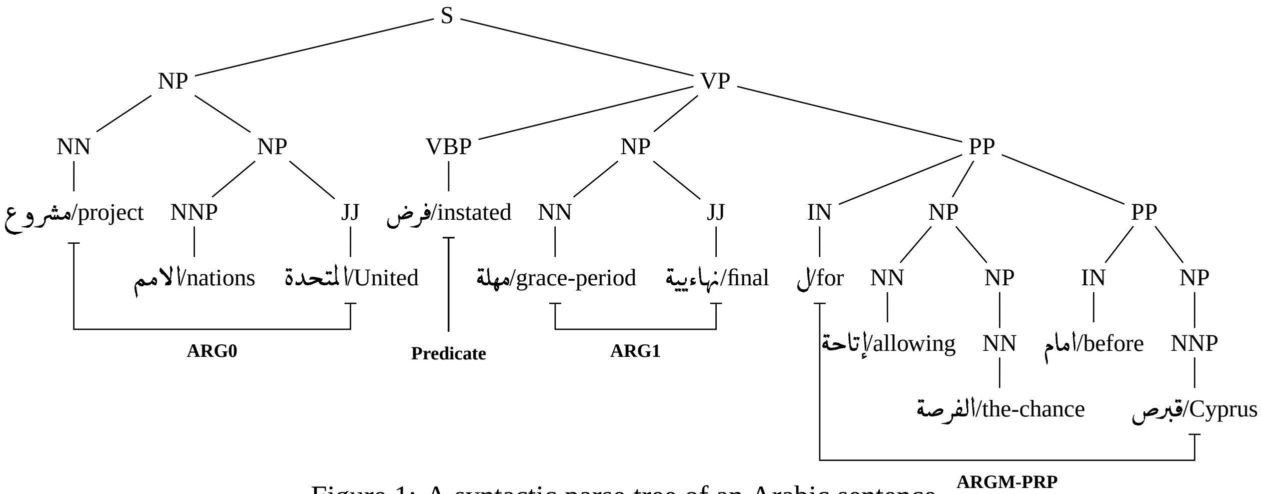

Figure 1: A syntactic parse tree of an Arabic sentence.

PropBank or *Agent* and *Goal* in FrameNet.

The standard approach to learn both the detection and the classification of predicate arguments is summarized by the following steps:

1. Given a sentence from the *training-set*, generate a full syntactic parse-tree;
2. let $\mathcal{P}$ and $\mathcal{A}$ be the set of predicates and the set of parse-tree nodes (i.e. the potential arguments), respectively;
3. for each pair $\langle p, a \rangle \in \mathcal{P} \times \mathcal{A}$:
   - extract the feature representation set, $F_{p,a}$;
   - if the subtree rooted in $a$ covers exactly the words of one argument of $p$, put $F_{p,a}$ in $T^+$ (positive examples), otherwise put it in $T^-$ (negative examples).

For instance, in Figure 1, for each combination of the predicate *instated* with the nodes `NP`, `S`, `VP`, `VPB`, `NNP`, `NN`, `PP`, `JJ` or `IN` the instances $F_{instated,a}$ are generated. In case the node $a$ exactly covers "project nations United", "grace-period final" or "for allowing the chance before Cyprus", $F_{p,a}$ will be a positive instance otherwise it will be a negative one, e.g. $F_{instated,IN}$.

The $T^+$ and $T^-$ sets are used to train the boundary classifier. To train the multi-class classifier, $T^+$ can be reorganized as positive $T^+_{arg_i}$ and negative $T^-_{arg_i}$ examples for each argument $i$. In this way, an individual ONE-vs-ALL classifier for each argument $i$ can be trained. We adopted this solution, according to (Pradhan et al., 2003), since it is simple and effective. In the classification phase, given an unseen sentence, all its $F_{p,a}$ are generated and classified by each individual classifier $C_i$. The argument associated with the maximum among the scores provided by the individual classifiers is eventually selected.

The above approach assigns labels independently for the different arguments in the predicate argument structure. As a consequence the classifier output may generate overlapping arguments. Thus, to make the annotations globally consistent, we apply a disambiguating heuristic that selects only one argument among multiple overlapping arguments. The heuristic is based on the following steps:

- if more than two nodes are involved, i.e. a node $d$ and two or more of its descendants $n_i$ are classified as arguments, then assume that $d$ is not an argument. This choice is justified by previous studies (Moschitti et al., 2005) showing that for lower nodes, the role classification is generally more accurate than for upper ones;
- if only two nodes are involved, i.e. they dominate each other, then keep the one with the higher SVM classification score.

### 2.2 Standard Features

The discovery of relevant features is, as usual, a complex task. However, there is a common consensus on the set of basic features that should be adopted. Among them, we select the following subset: (a) *Phrase Type*, *Predicate Word*, *Head Word*, *Position* and *Voice* as defined in (Gildea and Jurafsky, 2002); (b) *Partial Path, No Direction Path, Head Word POS, First and Last Word/POS in Constituent* and *SubCategorization* as proposed in (Pradhan et al., 2003); and (c) *Syntactic Frame* as designed in (Xue and Palmer, 2004).

For example, *Phrase Type* indicates the syntactic type of the phrase labeled as a predicate argument, NP for *Arg1* in Figure 1 whereas the *Parse Tree Path* contains the path in the parse tree between the predicate and the argument phrase, expressed as a sequence of nonterminal labels linked by direction (up or down) symbols, VPB ↑ VP ↑ S ↓ NP for *Arg1* in Figure 1.

## 3 Experiments

In these experiments, we investigate if the technology proposed in previous work for automatic SRL of English texts is suitable for Arabic SRL systems. From this perspective, we tested each SRL phase, i.e. boundary detection and argument classification, separately.

The final labeling accuracy that we derive using the official CoNLL evaluator (Carreras and Màrquez, 2005) along with the official development and test data of SEMEVAL provides a reliable assessment of the accuracy achievable by our SRL model.

### 3.1 Experimental setup

We use the dataset released in the SEMEVAL 2007 Task 18 on Arabic Semantic Labeling, which is sampled from the Pilot Arabic PropBank. Such data covers the 95 most frequent verbs in the Arabic Treebank III ver. 2 (ATB) (Maamouri et al., 2004). The ATB consists of MSA newswire data from Annhar newspaper from the months of July through November 2002.

An important characteristic of the dataset is the use of unvowelized Arabic in the Buckwalter transliteration scheme. We used the gold standard parses in the ATB as a source for syntactic parses for the data. The data comprises a development set of 886 sentences, a test set of 902 sentences, and a training set of 8,402 sentences. The development set comprises 1,725 argument instances, the test data comprises 1,661 argument instances, and training data comprises 21,194 argument instances. These instances are distributed over 26 different role types.

|  | Precision | Recall | $F_{\beta=1}$ |
|---|---|---|---|
| Dev | 97.85% | 89.86% | 93.68 |
| Test | 97.85% | 90.55% | 94.06 |

Table 1: Boundary detection F1 results on the development and test sets.

The training instances for the boundary detection task relate to parse-tree nodes that do not correspond to correct boundaries. For efficiency reasons, we use only the first 350K training instances for the boundary classifier out of more than 700K available.

The experiments are carried out with the SVM-light-TK software available at http://ai-nlp.info.uniroma2.it/moschitti/ which encodes tree kernels in the SVM-light software. This allows us to design a system which can exploit tree kernels in future research. To implement the boundary classifier and the individual argument classifiers, we use a polynomial kernel with the default regularization parameter (of SVM-light), and a cost-factor equal to 1.

### 3.2 Official System Results

Our system is evaluated using the official CoNLL evaluator (Carreras and Màrquez, 2005), available at http://www.lsi.upc.es/∼srlconll/soft.html.

Table 1 shows the F1 scores obtained on the development and test data. We note that the F1 on the development set, i.e. 93.68, is slightly lower than the result on the test set, i.e. 94.06. This suggests that the test data is *easier* than the development set.

Similar behavior can be observed for the role classification task in tables[1] 2 and 3.

Again, the overall F1 on the development set (77.85) is lower than the result on the test set (81.43). This confirms that the test data is, indeed, *easier* than the development set.

Regarding the F1 of individual arguments, we note that, as for English SRL, ARG0 shows high values, 95.42 and 96.69 on the development and test sets, respectively. Interestingly, ARG1 seems

[1] The arguments: ARG1-PRD, ARG2-STR, ARG4, ARGM, ARGM-BNF, ARGM-DIR, ARGM-DIS, ARGM-EXT and ARGM-REC have F1 equal to 0. To save space, we removed them from the tables, but their presence makes the classification task more complex than if they were removed from test data.

| | Precision | Recall | $F_{\beta=1}$ |
|---|---|---|---|
| Overall | 81.31% | 74.67% | 77.85 |
| ARG0 | 94.40% | 96.48% | 95.42 |
| ARG1 | 91.69% | 88.03% | 89.83 |
| ARG1-PRD | 50.00% | 50.00% | 50.00 |
| ARG1-STR | 20.00% | 4.35% | 7.14 |
| ARG2 | 60.51% | 61.78% | 61.14 |
| ARG3 | 66.67% | 15.38% | 25.00 |
| ARGM | 100.00% | 16.67% | 28.57 |
| ARGM-ADV | 46.39% | 43.69% | 45.00 |
| ARGM-CND | 66.67% | 33.33% | 44.44 |
| ARGM-DIS | 60.00% | 37.50% | 46.15 |
| ARGM-LOC | 69.00% | 84.15% | 75.82 |
| ARGM-MNR | 63.08% | 48.24% | 54.67 |
| ARGM-NEG | 87.06% | 97.37% | 91.93 |
| ARGM-PRD | 25.00% | 7.14% | 11.11 |
| ARGM-PRP | 85.29% | 69.05% | 76.32 |
| ARGM-TMP | 82.05% | 66.67% | 73.56 |

Table 2: Argument classification results on the development set.

| | Precision | Recall | $F_{\beta=1}$ |
|---|---|---|---|
| Overall | 84.71% | 78.39% | 81.43 |
| ARG0 | 96.50% | 96.88% | 96.69 |
| ARG0-STR | 100.00% | 20.00% | 33.33 |
| ARG1 | 92.06% | 89.56% | 90.79 |
| ARG1-STR | 33.33% | 15.38% | 21.05 |
| ARG2 | 70.74% | 73.89% | 72.28 |
| ARG3 | 50.00% | 8.33% | 14.29 |
| ARGM-ADV | 64.29% | 54.78% | 59.15 |
| ARGM-CAU | 100.00% | 9.09% | 16.67 |
| ARGM-CND | 25.00% | 33.33% | 28.57 |
| ARGM-LOC | 67.50% | 88.52% | 76.60 |
| ARGM-MNR | 54.17% | 47.27% | 50.49 |
| ARGM-NEG | 80.85% | 97.44% | 88.37 |
| ARGM-PRD | 20.00% | 8.33% | 11.76 |
| ARGM-PRP | 85.71% | 66.67% | 75.00 |
| ARGM-TMP | 90.82% | 83.18% | 86.83 |

Table 3: Argument classification results on the test set.

more difficult classify in Arabic than it is in English. In our current experiments, the F1 for ARG1 is only 89.83 (compared to 95.42 for ARG0). This may be attributed to two main factors. Arabic allows for different types of syntactic configurations, subject-verb-object, object-verb-subject, verb-subject-object, hence the logical object of a predicate is highly confusable with the logical subject. Moreover, around 30% of the ATB data is pro-dropped, where the subject is morphologically marked on the verb and its absence is marked in the gold trees with an empty trace. In the current version of the data, the traces are annotated with the ARG0 semantic role consistently allowing for the high relative performance yielded.

The F1 of the other arguments seems to follow the English SRL behavior as their lower value depends on the lower number of available training examples.

## 4 Conclusion

In this paper, we presented a first system for Arabic SRL system. The system yields results that are very promising, 94.06 for argument boundary detection and 81.43 on argument classification.

For future work, we would like to experiment with explicit morphological features and different POS tag sets that are tailored to Arabic. The results presented here are based on gold parses. We would like to experiment with automatic parses and shallower representations such as chunked data. Finally, we would like to experiment with more sophisticated kernels, the tree kernels described in (Moschitti, 2004), i.e. models that have shown a lot of promise for the English SRL process.

## Acknowledgements

The first author is funded by DARPA Contract No. HR0011-06-C-0023.

## References

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of CoNLL-2005*, Ann Arbor, Michigan.

Katrin Erk and Sebastian Pado. 2006. Shalmaneser - a flexible toolbox for semantic role assignment. In *Proceedings of LREC-06*, Genoa, Italy.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wig dan Mekki. 2004. The Penn-Arabic Treebank : Building a large-scale annotated Arabic corpus.

Alessandro Moschitti, Ana-Maria Giuglea, Bonaventura Coppola, and Roberto Basili. 2005. Hierarchical semantic role labeling. In *Proceedings of CoNLL-2005*, Ann Arbor, Michigan.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *proceedings of ACL-2004*, Barcelona, Spain.

Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2003. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of ICDM-2003*, Melbourne, USA.

Honglin Sun and Daniel Jurafsky. 2004. Shallow semantic parsing of chinese. In *In Proceedings of NAACL 2004*, Boston, USA.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP 2004*, pages 88–94, Barcelona, Spain.

Nianwen Xue and Martha Palmer. 2005. Automatic semantic role labeling for chinese verbs. In *Proceedings of IJCAI*, Edinburgh, Scotland.