Syntactic analysis of the Tunisian Arabic

Asma Mekki, Inès Zribi, Mariem Ellouze, Lamia Hadrich Belguith

ANLP Research Group, MIRACL Lab., University of Sfax, Tunisia
asma.elmekki.ec@gmail.com, ineszribi@gmail.com,
mariem.ellouze@planet.tn, l.belguith@fsegs.rnu.tn

Abstract. In this paper, we study the problem of syntactic analysis of Dialectal Arabic (DA). Actually, corpora are considered as an important resource for the automatic processing of languages. Thus, we propose a method of creating a treebank for the Tunisian Arabic (TA) "*Tunisian Treebank*" in order to adapt an Arabic parser to treat the TA which is considered as a variant of the Arabic language.

Keywords: Dialectal Arabic, Syntactic analysis, Treebank creation, Tunisian Arabic.

1 Introduction

Arabic language is a mixture of the Dialectal Arabic (DA) used by ~~the~~ Arabian native speakers and Modern Standard Arabic (MSA), the official language studied in schools, newspapers, etc. Nowadays, Arabic Dialects (AD) are the most widely used variety of Arabic, which promotes their treatments. However, the dialects mark phonological, morphological, syntactic and lexical differences when compared to the MSA. Actually, AD written in social networks, blogs or even some written documents do not follow an orthographic standard, which complicates the fact of having some adequate corpora able to be used for creating linguistic tools for DA.

In this paper, we will propose a method for creating a syntactic parser in favor of the Tunisian Arabic. First, we will identify the syntactic differences between the MSA and the TA. Then, we will present an overview of the Arabic syntactic parsers. Next, we will detail our proposed method for the creation of a treebank for the TA baptized "*Tunisian Treebank*". In section 5, we will present the adaptation of Stanford Parser to the TA. Finally, we will propose an evaluation of our method and conclude with perspectives.

2 Syntactic differences MSA–TA

The structuring of the MSA is among its main characteristics. Indeed, this one is well assured thanks to a grammar rich with universally recognized rules to follow, which is not the case for Arabic dialects. Indeed, the syntax of DA is affected by the influence of foreign languages and by the code switching between DA and MSA and even with

foreign languages. In addition, in the order of the words of the sentence, the nominal sentences are constituted syntactically of a subject and a predicate. For example, the nominal sentence المكان جميل *AlmakAn jamylN* "the place is beautiful" can be pronounced in TA in two ways either البلاصة مزيانة *AlblASaħ mizyanaħ* or مزيانة البلاصة mizyanaħ AlblASaħ. In addition, the inversion of the order between the MSA and the TA in several nominal groups is preferable.

3 Related Works

3.1 Parsers for MSA

Berkeley parser. Berkeley parser [1] uses a split-merge algorithm to learn a constituent grammar started with an x-bar grammar. Indeed, the split provides a tight fit to the training data, while the merge improves generalization and controls the size of the grammar. This analyzer is available in open source for other languages such as English, German, Chinese, etc. but not for Arabic.

Stanford parser. Stanford parser [2], created by Green and Manning is a grammar-based analyzer. It uses the non-contextual stochastic grammars to solve the syntactic analysis. It was trained on Arabic Treebank [2]. This parser does not contain an integrated tokenization tool, so the corpus to be used must already be tokenized in order to segment (clitic pronouns, prepositions, conjunctions, etc.). However, this parser does not require the segmentation of clitic determinants ال *Al* "*the*". The Stanford parser [2] is available in open source.

3.2 Parsers for Dialectal Arabic

Maamouri et al. [3] presented a syntactic analysis method that does not require an annotated corpora for DA (except for development and testing), or a parallel MSA/LEV[1] corpora. On the other hand, it requires the presence of a lexicon linking the DA lexemes to the MSA lexemes and the knowledge of the morphological and syntactic differences between the MSA and a dialect.
Three methods have been proposed for the syntactic analysis of DA [4]: the transduction of sentences, as well as that of the treebank and also the transduction of grammar. The basic idea of sentences transduction method is to translate the words of a sentence of the DA into one or more words in MSA that will be kept in the form of trellis. The best path in the lattice is transmitted to the MSA analyzer [5]. Finally, they replace the terminal nodes of the resulting analysis structure with the original words in the LEV dialect. The second method is the treebank transduction. The basic idea is to convert the MSA treebank (ATB), in an approximation, into a treebank for DA using the linguistic knowledge of systematic syntactic, lexical and morphological variations between the two varieties of Arabic. On this new treebank, the syntactic

[1] Levantine Arabic.

parser of [5] is learned and then evaluated on the LEV. Finally, grammatical transduction method encompasses the other two methods [4]. It uses the synchronous grammar mechanism to generate tree pairs linking the syntactic structures of the MSA and LEV sentences. These synchronous grammars can be used to analyze new dialect phrases. The evaluation of these three methods showed that the transduction of the grammar gave the best performance. It reduced the error rate by 10.4% and 15.3% respectively, with and without the use of grammatical category labels.

4 "*Tunisian Treebank*" creation

The following section details the steps of our method for creating a treebank for the TA: "Tunisian Treebank". Figure 1 shows the step of "*Tunisian Treebank*" creation.

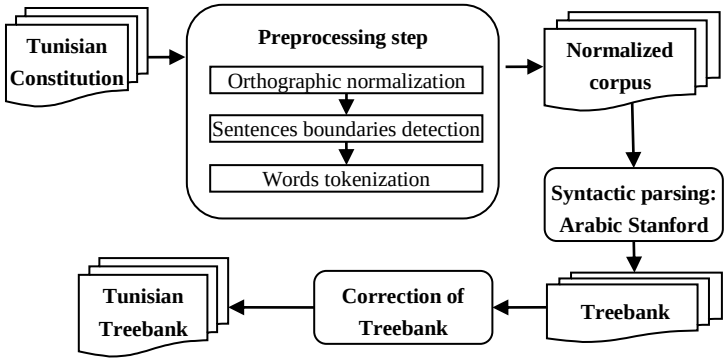


Fig. 1. Steps of "*Tunisian Treebank*" creation

4.1 Presentation of the "Tunisian Constitution"

After the events of the Tunisian revolution, a version of the Tunisian constitution was elaborated in TA in order to make it more comprehensible. Indeed, this corpus is one of the rare examples of intellectualized dialect directly written in TA. It contains 12 k words distributed among 492 sentences.

4.2 Preprocessing step

Before starting the syntactic analysis of the TA, the "*Tunisian constitution*" must go through a preprocessing step. It consists of orthographic normalization, sentence boundaries detection and words tokenization of the corpus.

Orthographic Normalization. The resource we have does not follow any orthographic convention. Consequently, we find that sometimes the same word is written in several ways in the corpus, which increases the ambiguity of our task. To do this,

the constitution must go through a normalization step to follow the orthographic convention of the TA "CODA-TUN" [6]. It defines a single orthographic interpretation for each word. Indeed, it follows the objectives and principles of work of the CODA [7]. Therefore, it is an internal coherence convention for the TA writing, which uses the Arabic alphabet and aims to find an optimal balance between maintaining a dialectal level of uniqueness and establishing conventions based on similarity MSA-TA. This normalization facilitates the adaptation of the syntactic parser of the MSA in favor of the TA. Thus, the number of modifications and treatments made during adaptation is reduced because of the sharing of several characteristics (word segmentation rules, derivation, etc.).

In this framework, we used the tool developed by [8] to perform the normalization in an automatic way. This step allowed us to unify the orthographic interpretation of each word of the constituent. For example, following the spelling convention of the TA "CODA-TUN" words such as *br$p* “many” and *vmp* “there are” are transcribed as برشة and ثمة.

Sentences boundaries detection. Sentences of the “*Tunisian constitution*” are not well segmented. Moreover, we find a two-page part called التوطئة or “*preface*”" without any point to delimit the sentences, which considerably complicates the phase of the syntactic analysis. Indeed, we will try to correct the segmentation of the sentences while maintaining their significance. To do this, two different experts participated in the manual correction of the segmentation.

Indeed, in the beginning, the resource consisted of 492 sentences but after experts segmentation correction, the number of sentences increased to 928. The maximum length of these sentences is 70 words and the minimum length is 2 words.

Words Tokenization. After the standardization of the “*Tunisian constitution*” according to the “CODA-TUN” convention and the segmentation of its sentences, we proceed to the words tokenization step. Indeed, Stanford Parser requires a tokenized entry, which implies the importance of this step. In fact, tokenization consists in defining the boundaries of the words and the information about the tokens that compose them (stem and clitics) [9].

4.3 Syntactic parsing: Stanford Parser.

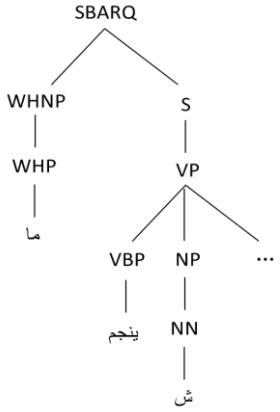
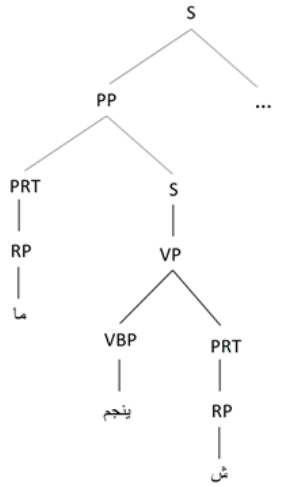
In order to create a treebank from the “*Tunisian constitution*”, the syntactic analyzer "Stanford Parser" will receive as input the sentences of the normalized constitution. The system dedicated primarily to MSA will give as an output a syntactic tree suitable for each sentence. In fact, this tree represents the structure of the sentence. Nevertheless, this output is not always admissible given the errors we can derive from it. In addition, it is worth noticing the differences, however limited, between the MSA and the TA, but the output presents errors that we must correct.

4.4 Correction of Treebank.

The last part of the creation of "Tunisian Treebank" is dedicated to the correction of parsing errors conducted by the "Stanford Parser" system. we can find mainly two types of errors: those that arise from structures specific to the TA and those whose words are not recognized by the system. Both Tables 1 and Table 2 below illustrate some examples of errors committed by the parser as well as the reference annotation.

In the first table, the negation structure of the sentence is not well-presented in its syntactic tree. Obviously, for this example a parent label to highlight this structure, which is not the case in this example, must enclose two negation particles as well as the verb. On the contrary, we note that the verb as well as the second particle of negation ش $ are joined with the remaining part of the sentence represented in the first example of Table 1 by "…". Moreover, labels used are erroneous since, in this case, the particle of negation ما *mA* is not an interrogative particle and the negation particle ش *$* is not a name.

Table 1. Example of structural errors committed by the parser.

Example	Syntactic tree	
	System	Reference
ما ينجم ش "He cannot"		

In Table 2, we present the annotation of two words not recognized by Stanford parser as well as the reference annotation.

Table 2. Examples of unrecognized words by Stanford parser.

Examples	Syntactic tree	
	System	Reference
علاش “why”	NP NNP علاش	WHADVP WHRB علاش
باش “In order to”	NP NNP باش	PRT RP باش

“Stanford Parser” typically uses the NNP label to annotate proper nouns as well as unrecognized words. Hence, “Stanford parser has attributed these labels to the two examples presented in Table 2.

We have prepared some statistics concerning the number of errors for each sentence of the corpus. Table 3 presents these statistics.

Table 3. Statistics classifying the sentences of the corpus according to the number of errors.

Syntactic error	**Number of words**		**Number of sentences**	**Percentage**
0	Min	3	129	13.90%
	Max	39		
1	Min	2	193	20.80%
	Max	31		
2	Min	5	172	18.53%
	Max	36		
3	Min	5	188	20.26%
	Max	43		
>=4	Min	6	246	26.51%
	Max	70		
Total			**928**	**100%**

The sentences of our corpus are classified according to the number of errors for each sentence. Therefore, we find that more than one third (1/3) of the corpus sentences either contain no fault or one fault that is usually due to a word not recognized by the system. This favors the idea of adapting the Arabic version of the Stanford Parser to the TA.

For correcting the generated treebank, we referred to two experts to help us annotate TA specific words and structures. These experts have corrected the annotation of the treebank to ensure the homogeneity of the treebank.

5 Adaptation of the system

The syntactic differences between the TA and the MSA are very limited, which favors the adaptation of a system dedicated to the MSA in order to generate the most appropriate model following the training phase. Indeed, we used the corpus that we created to do the training. This phase allows the system to generate a model able to give the labels of each word in its context and define the best hierarchical structure to use. Thus, the generated model has given results that we will try to improve by setting the highlighting attributes.

6 Evaluation

In this section, we use the cross-validation method, which represents the reliability estimate of our model. In fact, we have opted for this method of validation since the treebank we have created contains only 928 syntactic trees. However, this number is not enough to divide into one part for learning and another part for testing the model.

Table 4 below shows recall, precision and F-measure values which are calculated according to a 10-fold cross-validation.

Table 4. The F-measure value following the adaptation of the Stanford Parser system

Model	Recall	Precision	F-measure
Tunisian Stanford Parser	65.7	63.29	64.47

We give in Table 4 the results for the evaluation measure Evalb which is a Java reimplementation indicating the accuracy, recall and F-measure for the corpus data. The results presented are calculated with the PCFG or non-contextual grammar with probabilities assigned to the rules so that the sum of all probabilities for all rules extending the same non-terminal equals one. This PCFG is incorporated into Stanford Parser.

Subsequently, we tried to improve the F-measure value obtained by parameterizing the attributes of which we detail the results found in Table 5. We note that these results are obtained following a learning phase with a 75% part of the treebank and the generated model was tested by the evaluation corpus, which represents 25% of the treebank. In this part, the partition of the treebank was not random, but in fact, taking a quarter of each part of the classification we had by performing as a criterion the number of errors in each sentence.

Table 5. The incremental values of F-measure following the parameterization of the Stanford Parser system

Attribute	F-measure	Improvement: F-measure
-	66.06	-
noNormalization	66.31	+0.25
useUnknownWordSignatures	67.5	+1.19
smartMutation	67.7	+0.39

The meaning of these attributes is descaled as follows:

- No Normalization: Used to normalize the syntax trees of our treebank. In fact, this option has been added in order to standardize the Penn Arabic Treebank (ATB), which has been annotated separately from the beginning, but since our treebank is based on the annotation of the Sanford Parser system then it would better disable it.
- Use Unknown Word Signatures: Applied to use the suffix and capitalization information for unknown words. Indeed, the values from 6 to 9 are the options dedicated to Arabic.
- Smart Mutation: Dedicated generally to promote a more intelligent smoothing for the relatively rare words in the corpus. Table 6 shows the last recall, precision and F-measure values that are calculated according to a 10-fold cross-validation.

Table 6. A comparison of the F-measure values before and after the parameterization of the Stanford Parser system.

Model		Recall	Precision	F-measure
Tunisian Stanford Parser	Before	65.7	63.29	64.47
	After	**66.77**	**64.43**	**65.58**

Table 6 shows the best result that we had following the parameter setting of the attributes. In fact, the value of F-measure increased by 1.11%. Obviously, this result is encouraging, but given the size of our treebank, this result can be significantly improved if we increase its size.

7 Conclusion

We presented a method of creating a treebank for the intellectualized TA from the Tunisian constitution. Indeed, we started to preprocess our corpus in order to normalize it by following the spelling convention “CODA-TUN”. Then, the constitution went through a stage of segmentation of the sentences in order to correct the segmentation of this corpus, since it presents very long sentences. Then, we completed the pre-processing step by tokenizing the constitution. Subsequently, the pretreated corpus went through the Arabic version of the syntactic parser “Stanford Parser” to out-

put a treebank “Tunisian Treebank” which we corrected. Since the TA is a variant of standard Arabic, we proposed to make the adaptation of the syntactic parser “Stanford Parser” to generate a model that we evaluated by our treebank. As a perspective, we opt to implement a tokenization tool for the TA, which will facilitate the analysis of this dialect. Similarly, we propose to increase the size of the treebank in order to greatly improve the results obtained.

8 References

1. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. Proc. 21st Int. Conf. Comput. Linguist. 44th Annu. Meet. Assoc. Comput. Linguist. 433–440 (2006).
2. Green, S., Manning, C.: Better Arabic parsing: Baselines, evaluations, and analysis. COLING ’10 Proc. 23rd Int. Conf. Comput. Linguist. 394–402 (2010).
3. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The penn arabic treebank: Building a large-scale annotated arabic corpus. NEMLAR Conf. Arab. Lang. Resour. Tools. 102–109 (2004).
4. Rambow, O., Chiang, D., Diab, M., Habash, N., Hwa, R., Sima’an, K., Lacey, V., Levy, R., Nichols, C., Shareef, S.: Parsing Arabic Dialects. (2006).
5. Bikel, D.M.: Design of a multi-lingual, parallel-processing statistical parsing engine. In: Proceedings of the second international conference on Human Language Technology Research. pp. 178–182 (2002).
6. Zribi, I., Boujelbane, R., Masmoudi, A., Ellouze, M., Belguith, L., Habash, N.: A Conventional Orthography for Tunisian Arabic. In: The Ninth International Conference on Language Resources and Evaluation (LREC’14). pp. 2355–2361 (2014).
7. Habash, N., Diab, M., Rambow, O.: Conventional Orthography for Dialectal Arabic. In: Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12). pp. 711–718. European Language Resources Association (ELRA), Istanbul, Turkey (2012).
8. Boujelbane, R., Zribi, I., Kharroubi, S., Ellouze, M.: An Automatic Process for Tunisian Arabic Orthography Normalization. In: Lecture Note in Computer Science, LNCS, S. (ed.) HrTAL 2016. , Dubrovnik, Croatia (2016).
9. Attia, M. a.: Arabic Tokenization System. Proc. 5th Work. Important Unresolved Matters. 65–72 (2009).