Exploring the Potential of Schemes in Building NLP Tools for Arabic Language

Mohamed Achraf Ben Mohamed, Souheyl Mallat, Mohamed Amine Nahdi, and Mounir Zrigui
LaTICE Laboratory, Faculty of Sciences of Monastir, Tunisia

Abstract: *Arabic is known for its sparseness, which explains the difficulty of its automatic processing. The Arabic language is based on schemes; lemmas are produced using derivation based on roots and schemes. This latter character presents two major advantages: First, this "hidden side" of the Arabic language composed of schemes suffers much less from sparseness since it represents a finite set, second, schemes keep a large number of features of the language in a much reduced vocabulary size. Schemes present a very great perspective and have great potential in building accurate natural language processing tools for Arabic. In this work we tried to explore this potential by building some NLP tools while relying entirely on schemes. The work is related to text classification and a Probabilistic Context Free Grammar (PCFG) parsing.*

Keywords: *Arabic language, schemes, roots, derivation, text classification, PCFG, parsing.*

Received August 18, 2013; accepted May 10, 2014; published online December 3, 2014

1. Introduction

Unlike other languages, Arabic language possesses an internal structure formed by schemes [2]. Schemes are kinds of templates that guide the production of nouns and verbs. Each noun or verb is obtained by "molding" a root, composed mainly of three letters, using a scheme [12, 22]. This mechanism may involve elongation, repetition or even adding characters (such as suffixes, prefixes and infixes) [13]. This hidden side of Arabic actively contributes to the synthesis and analysis of this language. The study of schemes is essential to develop natural language processing tools for Arabic. One of the advantages of schemes is that they do not suffer from the sparse characteristic of Arabic which always regarded as an obstacle to the development of NLP systems for this language [14, 16]. The present work highlights this characteristic by redefining some classical NLP concepts while taking into account the use of schemes. Using schemes instead of plain text can be seen as an abstraction of the Arabic language aiming to retain only features that can be relevant for automatic processing of Arabic. In this work, we try to explore the advantages and limitations of this approach. We explored the use of schemes in two fields:

- Text Classification: By building a neural network classifier based on schemes.
- Parsing: By building a Probabilistic Context Free Grammar (PCFG) parser based entirely on schemes.

1.1. Related Work

There has been a lot of interest in using schemes in morphosyntactic analysis for Arabic especially for stemming [13]. However, these approaches limit the use of schemes at the level of word. It's in this fact that led us to explore the use of schemes in larger scale by producing NLP tools for Arabic while relying on schemes.

1.2. Arabic Language

The Arabic language is the language of the Koran, the sacred book of Muslims. It is a Semitic language. Arabic is written from right to left [8]. The Arabic alphabet has 28 consonants that change shape depending on presentation of their position in the word Table 1.

Table 1. Variation of the letter ع.

End (Unreachable Letter)	End	Middle	First
ع	ـع	ـعـ	عـ

Diacritics, originally nonexistent in Arabic [3, 4] are used to eliminate ambiguity [21]; indeed a word without diacritics could have multiple possible interpretations; the word كتب for example can mean:

- كَتَبَ (Kataba, he wrote).
- كُتِبَ (Kutiba, it was written).
- كُتُبٌ (Kutub, books).

Arabic Language is composed of verbs, nouns and particles [10, 11]. From an NLP point of view, Arabic language, like any other language possesses both positive and negative aspects.

1.2.1. Negative Aspects of the Arabic Language

Arabic is highly inflected and agglutinative language which explains its sparseness [23]. The single Arabic word أَنُلْزِمُكُمُوهَا (Anulzimukumuha) is translated into an entire English sentence "should we force it upon you". Also, Arabic language is mostly written without

diacritics [17, 21]; this causes ambiguity as shown earlier. These morphological and syntactic properties of the Arabic language make this language hard to process when compared with other languages [15, 18].

1.2.2. Positive Aspects of the Arabic Language

Arabic is based on derivation [2]; it's a kind of lemma production using roots and schemes (patterns). Most roots are trilateral like كتب or قرأ. The basic root is represented by the word (فعل). Then, using a set of schemes, a whole semantic concept will be generated from each root [14, 23]. There are two types of schemes (derivations): Verbal and nominal. Figure 1 shows an example of derivation of the root: كتب .

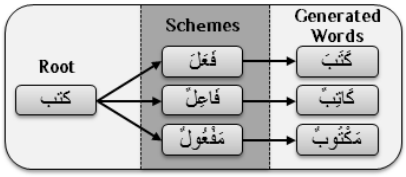


Figure 1. Example of derivation of the root كتب.

The ultimate objective of this study is to build NLP tools at the level of schemes by using schemes exclusively instead of plain text. In the following we will justify this choice by giving some characteristics of this "hidden side" of Arabic. Table 2 shows an example of schemes conversion.

Table 2. Example of schemes conversion.

Schemes	Text
...اسْتَأْنَافَ فَعَلَهُ فَاعِلٌ	...إِلَّا رَدَّهُ خَاطِرٌ
شرط جر فِعْلِهِ اسْتَأْنَافَ	إِمَّا مِنْ نَوْعِهِ أَوْ
جر جزم فِعْلِهِ اسْتَأْنَافَ	مِنْ غَيْرِ نَوْعِهِ ثُمَّ
الْفِعَالُ الْفَاعِلَةُ	الْمِيَاهُ الْجَارِيَةُ
جارومجرور شرط يَفْعَلْ	مِنْهَا مَا يَنْفَعُ
جارومجرور شرط يَفْعَلْ...	وَمِنْهَا مَا يَضُرُّ...

From an NLP point of view; conversion from text to schemes is characterize by a significant vocabulary size reduction. Figure 2 shows the result after the conversion of text of 100k words. Reduction for this case is equal to 92.90%.

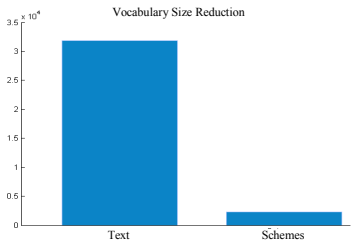


Figure 2. Vocabulary size reduction.

From a statistical point of view conversion from text to schemes goes with a reorganization of terms (words/ schemes) as exposed in Figure 3 which shows the frequency distributions for a text and for its conversion into schemes. This reorganization is characterized by two main transformations:

- Decrease of Number of Classes.
- Increase of the Cardinality of Classes.

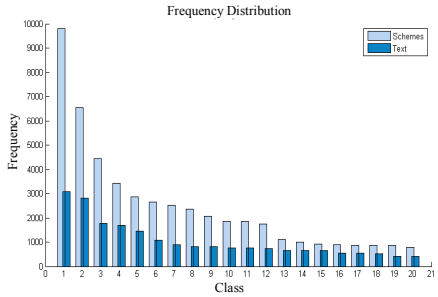


Figure 3. Frequency distributions.

Also, by calculating the standard deviation and the mean of the sample we can draw the normal distributions of both models in Figure 4.

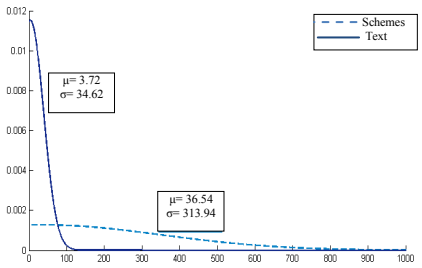


Figure 4. Normal distributions.

Again we can see that for plain text the maximum of density goes for little size classes and that for schemes the distribution is more uniform; similarly, we can have several class sizes.

1.2.3. Conclusions

Using schemes helps reduce the effects of model sparseness which implies that the level of schemes can have a great potential in building NLP tools for Arabic. It is in this perspective that we have chosen to propose dealing exclusively with schemes level which can be considered as the hidden side of the Arabic language and that can help minimizing the effects of its sparseness. In the following we will present two schemes based systems: Text classification and a PCFG parser.

2. Text Classification

2.1. Introduction

Text classification is the task of assigning a class to some piece of text. One of the most common methods for performing text classification is supervised machine learning [1, 23]. This method consists of providing the system, in addition to the document and the possible classes, a training set. It's a set of documents (texts) manually labeled. The system will use this set to identify the features of each class. Then, it will extract the features of the document and decide which class the document belongs to. In this first experiment, we propose to perform this task for Arabic while exploiting schemes.

2.2. Schemes Properties

Schemes contain meanings and can contribute to the disambiguation of an Arabic text [2]. Table 3 shows words generated using particular schemes.

Table 3. Words generated using particular schemes.

Scheme	فعالة		مفاعلة		فعيل	
Generated Words	تجارة	Trade	مصافحة	Handshake	صهيل	Neigh
	بقالة	Grocery	مقابلة	Meeting	نهيق	Braying
	نجارة	Carpentry	مكالمة	Call	أزيز	Wheeze
	حدادة	Blacksmith	مبادلة	Exchange	صرير	Creak
Meaning	Profession		Participation		Sound	

As we might notice these generated words belong, in most cases, to the same lexical field. This can justify the use of schemes in text classification.

2.3. The System

We started our experiment by choosing the Naïve Bayes Classifier (NBC). Then, we measured the average accuracy while feeding the training set by 10% every time. Figure 5 gives the obtained result.

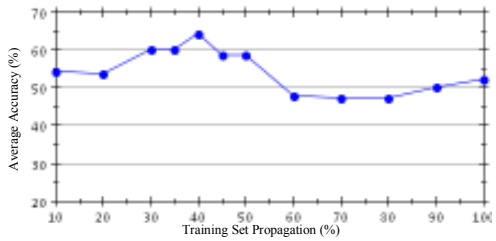


Figure 5. Schemes based NBC accuracy while varying training set size.

As we might notice that obtained result is far from classical classifier (less than 70% for ~40% of training set); nevertheless we will not exclude the use of schemes in text classification. Actually the problem was the use of NBC which is based on the hypothesis that terms are independent [6]. So, the solution was to use another classifier which did not take into account terms redundancy (dependence); the choice was neural networks. Figure 6 shows our system. In the following we will detail this neural networks schemes based text classifier.

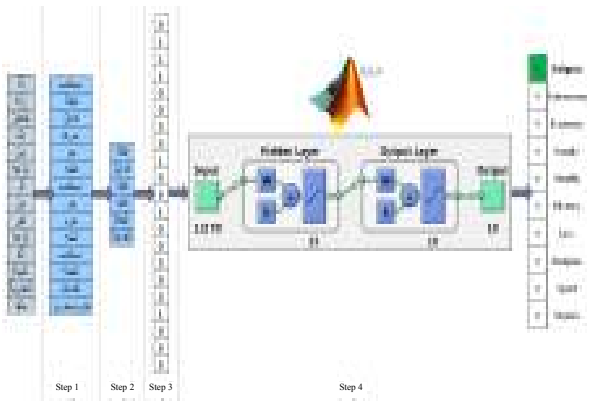


Figure 6. Neural networks schemes based text classifier.

To classify a text we start by converting it to schemes step 1, then we maintain only schemes having 'فعل' (verb) as root step 2. The next step is vectorization step 3. The network has 11370 input layers, 25 hidden layers and 10 output layers. To perform the test we used the Open Source Arabic corpus (OSAc) which contains 10 classes (religion, astronomy, economy, family, health, history, low, recipies, sport and stories). We divided OSAc corpus into three sets:

- 60% to train the network.
- 20% for validation. Figure 7 gives the optimal regularization parameter λ (λ in {0.01, 0.03, 0.1, 0.3, 1, 3}) and the number of iterations *m* (*m* in [1:50]).
- 20% for test.

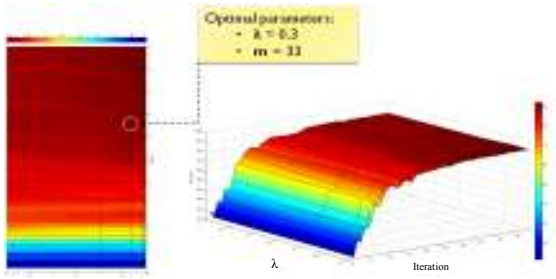


Figure 7. Neural network optimal parameters.

2.4. Result

Text classification by neural networks based on schemes has achieved an accuracy of 91.77%, which represents a significantly higher performance to that achieved by a Bayes classifier.

2.5. Conclusions

Schemes are a powerful tool for the classification of Arabic texts provided to select a tool not assuming independence between words.

3. Scheme Based PCFG Parser

3.1. Introduction

PCFG is a CFG with probabilities added to the rules; it represents the simplest and most natural probabilistic model for tree structures [7]. PCFG parsing cannot be easily used for Arabic language; this is due to the sparseness of the language [9, 19]. Building a PCFG for Arabic requires creating a rule base whose size is equal to the size of the vocabulary. Then, the base will be enriched by further rules allowing analyzing sentences with increasing complexity. The problem is that the use of a larger model exacerbates the sparse data problem [5]. Based on the fact that the schemes suffer much less from sparseness and have a very much reduced vocabulary size, we have built a PCFG at the level of schemes. Figure 8 illustrates this idea. We have an example of three sentences. The parsing of

these sentences using a classical PCFG parser would have required the rules base given in Table 4. The conversion into schemes of these three sentences gives the same result (الفَعَلُ فَاعِلٌ). The idea is to convert sentences into schemes and then parse the schemes sentences using rules written with schemes in Table 5. Finally the work is reduced to a correspondence between words and schemes.

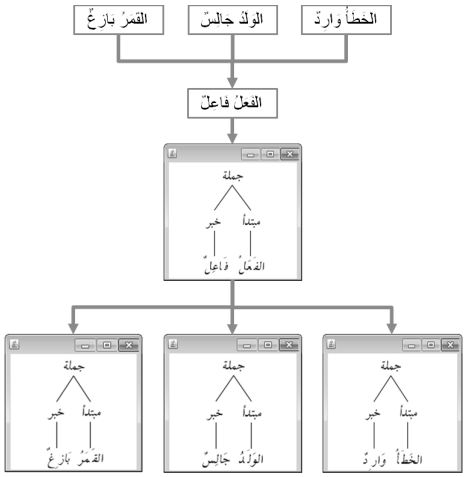


Figure 8. Example of schemes based PCFG parser.

Table 4. Rules base.

Rules base	
Rule	**Probability**
جملة ← مبتدأ خبر	1.0
مبتدأ ← الخَطأ	0.33
مبتدأ ← الوَلَدُ	0.33
مبتدأ ← القَمَرُ	0.33
خبر ← وَارِدٌ	0.33
خبر ← جَالِسٌ	0.33
خبر ← بَازِغٌ	0.33

Table 5. Schemes rules base.

Rules base	
Rule	**Probability**
جملة ← مبتدأ خبر	1.0
مبتدأ ← الفَعَلُ	1.0
خبر ← فَاعِلٌ	1.0

The use of schemes to write rules ensures a much broader coverage of the language using a less number of rules. In fact, each rule written with schemes represents not just a sentence (or a component of a sentence), but rather a pattern of sentence (or a component of a sentence).

3.2. The Grammar

Grammar is denoted by $G(T, N, H, S, R, P)$ where:

- T: Is a set of terminal symbols.
- N: Is a set of non-terminal symbols.
- H: Is a set of pseudo terminal symbols (schemes).
- S: Is a start symbol ($S \in N$).
- R: Is a set of rules: $A \rightarrow B$ where $A \in N \cup H$ and $B \in N \cup H \cup T$.
- P: Is a probability function:
- P: $R \rightarrow [0, 1]$.
- $\forall A \in N, \sum_{A \rightarrow B \in R} P(A \rightarrow B) = 1$

All rules are in Chomsky Normal Form (CNF) and are classified into two sets: Grammar and schemes. Where:

$$R_{Grammar} = \{A \rightarrow B/A, B \in N \vee (A \in N \wedge B \in T)\}$$

And

$$R_{Schemes} = \{A \rightarrow B/A \in N \wedge B \in H\}$$

The new parameter H represents the schemes and intervenes in the definition of rules. Schemes are not regarded as being terminal or non terminal that is why we chose to call them pseudo terminal.

3.3. Experiment

Figure 9 shows the different steps in parsing a sentence using schemes. In the following section we will detail each component of this diagram.

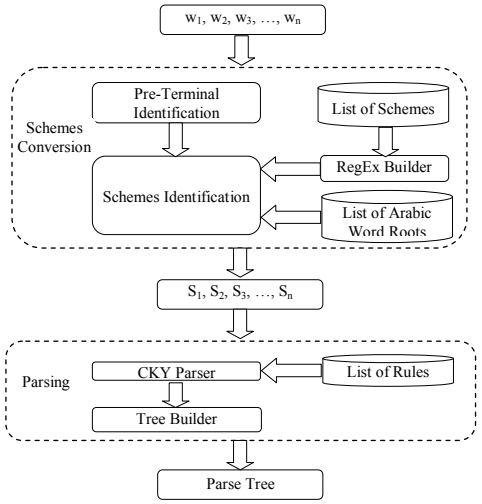


Figure 9. Schemes based PCFG parser diagram.

3.3.1. Schemes Conversion

The first step will be the conversion of the plain text into schemes. To convert a sentence, we begin first by recognizing the pre-terminals, such as:

- The Particles of Coordination (ثمّ، أو، أم، أمّا، إمّا…).
- The Interrogative Particles (أيّ، كيف، أين، متى…).
- The Particles of Appeal (يا، أيا، أي، هيا…).
- Prepositions (من، إلى، عن، على، ربّ، حتّى…).
- Conditional Particles (مهما، كيفما، حيثما…).
- Etc.

Then, we move to the recognition of schemes. We have built a base of schemes; we collected 3027 schemes, representing the most commonly used schemes, including 2031 verbal and 996 nominal. Schemes identification is performed by the regular expression. The system will produce for each scheme the corresponding regular expression which identifies which scheme the word in question corresponds to. Finally, a comparison with the list of Arabic words roots will avoid some conversion ambiguity as shown in Table 6.

Table 6. Schemes conversion ambiguity example.

Word	Possible Conversion	Correspondent Root	Decision
شارك	فعلك	شار	Rejected
	فاعل	شرك	Accepted

3.3.2. Tags Nomenclature System

To tag schemes we choose a special nomenclature system for showing grammatical features of each scheme. These features are very important in building rules. Intuitively, we have interest in extracting all grammatical features of a scheme. Unfortunately, we this was limited by computers processing capacity. We were forced to limit the number of features by keeping only the most important. In what follows, we will detail the system.

3.3.2.1. Nomenclature System for Verbs

For verbs, the most important feature is tense. A verbal scheme will have the following form: VerbalScheme_integer_; where integer indicates the tense. Examples:

- "_فعل ١_" indicates a verb in the past. Example: "تَفَاعَلْنَا".
- "_فعل 2_" indicates a verb in the present. Example: "نَتَفَاعَلُ".
- Etc.

3.3.2.2. Nomenclature System for Nouns

The most important characteristics for nominal schemes are category (اسم, فاعل, اسم مفعول), the last vowel, and the recognition. Recognized words in Arabic all begin with "ال". A nominal scheme will have the following form: NominalScheme_integer1_integer2_; where integer1 indicates the last diacritic and integer2 indicates the recognition state. Examples:

- "_اسم فاعل ٥ ٠_" indicates the category "فاعل اسم", with "ٌ" as last vowel and not recognized. Example: "مُفْتَعِلٌ".
- "_اسم_٢_١_" indicates the category "اسم", with "ُ" as last vowel and recognized. Example: "الفِعَالُ".
- Etc.

3.3.2.3. Nomenclature System for Unknown Words

Unknown words are words that system was unable to convert into schemes. It is quite frequent to come across this case, as for proper names or words partially vowelized. For such words, we chose the label "مجهول" (Unknown) and we adopted the same nomenclature system used for names. Such choice allows the system during the parsing process to "guess" the tag assigned based on the position of the word in the sentence, the last vowel and the recognition.

3.3.3. Agglutination

Agglutination is the major problem of Arabic; a word like (فَأَسْقَيْنَاكُمُوهُ) expresses a whole English sentence "And given you drink from it". This word is decomposed into several components: Enclitic, suffix, basic scheme, prefix and proclitic.

To solve this problem we used regular expressions again. Each basic scheme is identified and separated from enclitics and proclitics which are adjacent to it (if they exist). Each component is then treated as a terminal. For example the word "فَكَلَّمَنَا" which corresponds to the basic scheme "فَعَّلَ" will be exploded into three components: "فَ", "فَعَّلَ" and "نَا" and will be interpreted as three separated words.

3.4. Rules

To build the rules base we used the common grammatical rules of the Arabic language. Figure 10 shows an example of these rules.

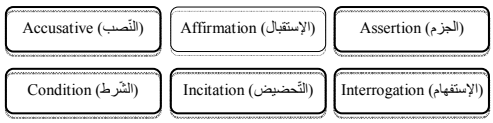


Figure 10. Characters and names that can be found in the beginning of the verbal sentence.

Rules were built in three levels: Atomic level, compounds level and phrase level where every level depends to the precedent one. In the following we will details these three levels.

3.4.1. Atomic Level

In this level, we are interested in lexicons. Each scheme can have only one label; excepting for unknown word which may have several possible functions.

Example of rules:

_فعل_١_ ← فَعَلَ
_اسمفاعل_٢_١_ ← المُفْعِلْ
_اسم_٢_٠_ ← _مجهول_٢_٠_
_اسممفعول_٢_٠_ ← _مجهول_٢_٠_

3.4.2. Compounds Level (المركّبات)

Compounds are a group of words (schemes) that can have a special function and that constitute a part of a sentence. It is at this level that appears the importance of the adopted nomenclature; indeed the schemes constituting each component must absolutely obey certain rules relating to the characteristics of this component. Example of compounds:

- Naati (نعتي) Example of Rule:

_منعوتنعت_١_ ← _منعوت_١_ _نعت_١_
_منعوت_١_ ← _اسم_١_١_
_نعت_١_ ← _اسمفاعل_١_١_

- Idhafi (إضافي) Example of Rule:

_مضافمضافإليه_١_ ← _مضاف_١_ _مضافإليه_١_
_مضاف_١_ ← _اسم_١_٠_
_مضافإليه_١_ ← _اسم_٣_١_

- Jarr (جرّ) Example of Rule:

جارمجرور ← _جار_ _مجرور_
جار ← _جار_
مجرور ← _اسممفعول_٣_١_

3.4.3. Phrase Level

This is the highest level in parsing. Arabic sentence is either verbal or nominal. Basically this level contains two rules:

جملة	←	مبتدأ	خبر
جملة	←	فعل	فاعل

Where the symbol «_جملة_» is the start symbol of the grammar.

Every category of sentence may have supplements; this implies that the number of rules governing this level is greater. We may have rules such:

جملة	←	-	+
-	←	_تحقيق_	_فعل_٢_
+	←	_فاعل_	_مفعولبه_

Where "+" and "-" are virtual non terminal used to respect the CNF.

3.5. Parsing

To find the most probable parse of the sentence according to our PCFG we used the Cocke-Younger-Kasami (CYK) algorithm [20]. This method gives us a cubic time algorithm: $\theta(n^3)$. Suppose we have to parse the following sentence: "فَكَلَّمَنَا سَعِيدٌ يَسْأَلُ عَنْ أَحْوَالِنَا" (Saiid contact us asking for our conditions). Tables 7 and 8 give respectively, the conversion into schemes and the most likely rules used to parse the sentence. Every part of the sentence is delimited by variables "Begin", "Span", "End" and is governed by a particular rule.

Table 7. Schemes conversion.

8	نَا / نَا	7	أَحْوَالِ / أَفْعَالِ	6	عَنْ / _جار_	5	يَسْأَلُ / يَفْعَلُ	4	سَعِيدٌ / فَعِيلٌ	3	نَا / نَا	2	كَلَّمَ / فَعَّلَ	1	فَ / فَ	0

Table 8. Most likely rules.

Rules				Begin	Span	End
.	-	←	_جملة_	0	2	8
_فعل_١_	_استأناف_	←	-	0	1	2
*	_مفعوليه_	←	.	2	3	8
حال	_فاعل_	←	*	3	4	8
جارمجرور	_فعل_٢_	←	_حال_	4	5	8
مجرور	_جار_	←	_جارمجرور_	5	6	8
ضمير	_اسم_٣_٠	←	_مجرور_	6	7	8

Figures 11 and 12 give the parse tree of this sentence respectively with and without taking into account virtual terminals.

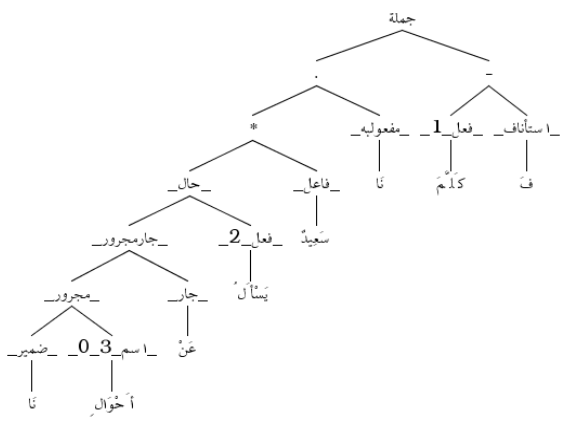


Figure 11. Parse tree with virtual non terminals.

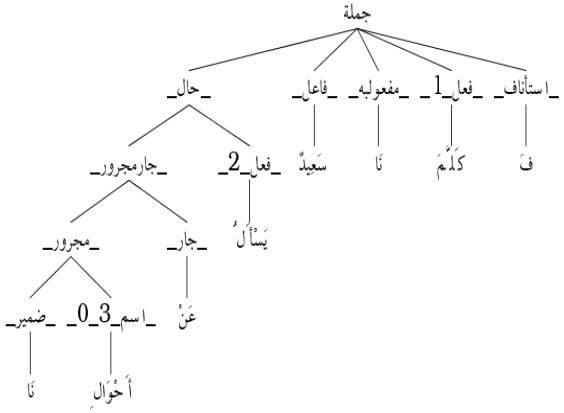


Figure 12. Parse tree without virtual non terminals.

3.6. Result

We used free and open online resources (news feeds, blogs, forums…) to extract and construct a test set. We created a test set of 836 sentences with an average length of 5.08 words. This provided us with accuracy of 63.35%. The accuracy was calculated based on the formula: Accuracy (%)=100*(Number of correctly tagged token)/(Total number of tokens). There are two main causes of failure in sentences parsing:

- Not all Sentences Categories are Covered by Our Grammar: It's obvious that we cannot build a rules base covering all Arabic language. There is a trade-off between the size of the rules base and the system execution time.
- The System Fail to Convert Some Words into Schemes: This may be due to incomplete or missing diacritization of words. This case occurs also with words having quadruple or quintuple roots not yet covered by our system.

3.7. Conclusions

During this work we built a PCFG parser with a grammar that deals with schemes instead of plain text. Each element of the rules base of this grammar actually replaces a very large number of rules we should have taken into account if we dealt with plain text. It's this compression, at the level of rules, which allowed building a PCFG for the Arabic language.

On the other hand, it is important to notice that in several cases where the system was unable to parse a sentence, the sentence components have been correctly guessed. This scenario occurred when the base of rules does not cover this particular category of sentences. So, as future work, we plan to make the system able to automatically enrich the base of rules by adding new rules. It is obvious that this process is very delicate and may require combining several NLP techniques.

In the same perspective, so far, the system uses the entire base of rules whatever the sentence being analyzed. This has an impact on the system execution time. In order to reduce the response time, we plan to make the base of rules dynamic; rules that cannot be part of the analysis will not be taken into account in the parsing process.

4. Overall Conclusions

In this work we tried to explore the potential of schemes in building NLP tools. We started a first exploration by the creation of a text classification system. The second task was the creation of a PCFG parser. The implementation of such system for Arabic is only feasible at the level of schemes since the creation of a PCFG parser for Arabic based on plain text will cover a limited part of the language. Whereas the use of schemes will cover a much larger part of the Arabic language since each scheme replaces all words that are derived from it.

The use of schemes on a larger scale has allowed mitigating the sparseness of Arabic and helped building more accurate NLP tools for this language.

Throughout our work, we tried to answer the following question: How far we can go with the use of schemes? The answer, in our opinion, is that it is always better, when it is possible, to make a total abstraction of the Arabic language by relying exclusively on schemes.

References

[1] Ayadi R., Maraoui M., and Zrigui M., "Intertextual Distance for Arabic Texts Classification," *in Proceedings of International Conference for Internet Technology and Secured Transactions*, London, UK, pp. 1-6, 2009.

[2] Badii E., *معجم الأوزان الصرفية (Glossary of Schemes)*. عالم الكتب للطباعة والنشر والتوزيع (World of books for printing, publishing and distribution), 1993.

[3] Ben Mohamed M., Ghoul D., Nahdi M., Mars M., and Zrigui M., "Arabic CALL System based on Pedagogically Indexed Text," *in Proceedings of International Conference on Artificial Intelligence*, Florida, USA, pp. 568-574, 2011.

[4] Ben Mohamed M., Zrigui M., and Maraoui M., "Clustering-Based Approach Extracting Collocations," available at: http://www.slideshare .net/mohamed achraf ben mohamed/clustering-based- approach- extracting-collocations, last visited 2013.

[5] Chen S. and Goodman J., "An Empirical Study of Smoothing Techniques for Language Modeling," *in Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, California, USA, pp. 310-318. 1996.

[6] Jakulin A., "Machine Learning based on Attribute Interactions," *PhD Thesis*, University of Ljubljana, 2005.

[7] Jurafsky D. and Martin H., *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition and Computational Linguistics*, Prentice-Hall, 2009.

[8] Khan B., Alghathbar K., Khan K., Alkelabi A., and Alajaji A., "Cyber Security using Arabic CAPTCHA Scheme," *the International Arab Journal of Information Technology*, vol. 10, no. 1, pp. 76-84, 2013.

[9] Manning D. and Schutze H., *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[10] Maraoui M., Antoniadis G., and Zrigui M., "CALL System for Arabic Based on Natural Language Processing Tools," *in Proceedings of the 4th Indian International Conference on Artificial Intelligence*, Tumkur, India, pp. 2249-2258, 2009.

[11] Maraoui M., "Elaboration D'un Dictionnaire Multifonction, a Large Couverture, De La Langue Arabe. Applications Aux Systèmes D'alao," *PhD Thesis*, Stendhal University, 2009.

[12] Mars M. "Analyse Morphologique Robuste De L'arabe et Applications Pédagogiques," *PhD Thesis*, Stendhal University, 2012.

[13] Marton Y., Habash N., and Rambow O., "Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features," *Computational Linguistics*, vol. 39, no. 1, pp. 161-194, 2013

[14] Meftouh K., Smaili K., and Laskri T., "Arabic Statistical Language Modeling," *in Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data*, Lyon, France, pp. 837-838, 2008.

[15] Merhbene L., Zouaghi A., and Zrigui M., "Ambiguous Arabic Words Disambiguation," *in Proceedings of the 11th International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing*, London, UK, pp. 157-164, 2010.

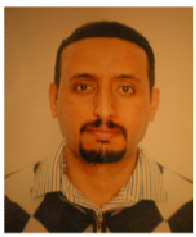
[16] Merhbene L., Zouaghi A., and Zrigui M., "Ambiguous Arabic Word Sense Disambiguation: the Results," available at: http://www. aclweb.org/anthology/R09-2009, last visited 2013.

[17] Motaz S. and Ashour W., "OSAC-Open Source Arabic Corpora," *in Proceedings of the 6th International Conference on Electrical and Computer Systems*, Lefke, North Cyprus, pp. 118-123, 2010.

[18] Saidane T., Zrigui M., and Ben Ahmed M., "Arabic Speech Synthesis using a Concatenation of Polyphones: The Results," *in Proceedings of the 18th Conference of the Canadian Society for Computational Studies of Intelligence*, Victoria, Canada, pp. 406-411, 2005.

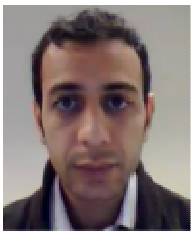
[19] Shaalan K., "Rule-based Approach in Arabic Natural Language Processing," *the International Journal on Information and Communication Technologies*, vol. 3, no. 3, pp. 11-19, 2010.

[20] Sikkel K. and Nijholt A., *Parsing of Context-Free Languages*, Springer Berlin Heidelberg, 1997.
[21] Zouaghi A., Merhbene L., and Zrigui M., "Combination of Information Retrieval Methods with LESK Algorithm for Arabic Word Sense Disambiguation," *Artificial Intelligence Review*, vol. 38, no. 4, pp. 257-269, 2012.
[22] Zouaghi A., Zrigui M., and Antoniadis G., "Automatic Understanding of Spontaneous Arabic Speech-A Numerical Model," *TAL*, vol. 49, no. 1, pp. 141-166, 2008.
[23] Zrigui M., Ayadi R., Mars M., and Maraoui M., "Arabic Text Classification Framework based on Latent Dirichlet Allocation," *Journal of Computing and Information Technology*, vol. 20, no. 2, pp. 125-140, 2012.



Mohamed Achraf Ben Mohamed is a PhD student in the Faculty of Economic Sciences and Management of Sfax, Tunisia. He is member of LaTICE Laboratory, Monastir unity (Tunisia). His areas of interest include natural language processing, computer-assisted language learning and machine learning.

Souheyl Mallat received his BCs degree in computer science from the Higher Institute of Applied Science and Technology of Sousse, Tunisia and his MSc degree from the Faculty of Sciences of Monastir, Tunisia. He is member of LaTICE Laboratory, Monastir unity (Tunisia). His areas of interest include natural language processing, data mining and information retrieval.



Mohamed Amine Nahdi received his BA degree in computer science at the Faculty of Sciences of Monastir, Tunisia and MA at the Grenoble Institute of Technology, France. He is a member of LATICE laboratory in Tunisia and LIDILEM laboratory in Grenoble France.

Mounir Zrigui is an associate professor at the University of Monastir, Tunisia. He received his PhD degree from the Paul Sabatier University, Toulouse, France in 1987 and his HDR in computer science from the Stendhal University, Grenoble, France in 2008. He has more than 25 years of experience including teaching and research in all aspects of automatic processing of natural language (written and oral).