The Correction of the Grammatical Case Endings Errors in Arabic Language

Chouaib MOUKRIM[1], Abderrahim TRAGHA[2], El Habib BENLAHMER[3]
Faculty of Sciences Ben M'sik, Hassan II University
Laboratory of Information Technologies and Modelling
Casablanca, Morocco

***Abstract*—Syntax plays a key role in natural language processing, but it does not always occupy an important position in applications. The main objective of this article is to solve the problem of the grammatical case ending errors produced by Arabic learners or certain common errors. Arabic can be considered more complex than English or French. He does not have vowels; diacritic signs (vowels) are placed above or below the letters. These diacritic signs are abandoned in most Arabic texts. This induces both grammatical and lexical ambiguities in Arabic. The present paper describes an automatic correction of this type of errors using "Stanford Parser" with an ontology containing the rules of the Arabic language. We segment the text into sentences, then we extract the annotations of each word with the syntactic relations coming from our parser, then we treat the relations obtained with our ontology. Finally, we compare the original sentence with the corrected one in order to detect the error. The implemented system achieved a total detection of about 94%. It is concluded that the approach is clearly promising by observing the results as compared to the limited number of available Arabic grammar checkers.**

Keywords*—Automatic correction; ontology; syntactic errors; case endings; natural language processing; Arabic

I. Introduction

Natural language processing (NLP) can appear as a homogenous and unified domain. Its goal is to design a software capable of automatically processing data expressed in a so-called "natural" language. These linguistic data may depend on the case, and are of different types (written texts, written or oral dialogues, etc.). It proposes a set of systems, allowing among others: to extract the information automatically, to summarize automatically long texts, to make an automatic translation of a text, and it also allows the generation of text, the parsing, etc.

Some NLP applications require syntactic analysis of texts apprehended in terms of grammatical relationships, and encoded as functional annotations in syntagmatic trees or dependency trees, which seems to be useful for many tasks. However, these parsers make errors in syntactic relationships and annotations; we can give an example with the following erroneous sentence: "دخـل الـمـعـلـمـون الـمـجـتـهـديـن"

دخل/VBD

المعلمون/DTNNS

المجتهدين/DTJJ

Root (Root, دخل)

dobj (المعلمون , دخل)

dep (المجتهدين ,المعلمون)

(S (VP (VBD دخل) (NP (DTNNS المعلمون) (DTJJ المجتهدين))))

Such as (see Table I)

We noticed that it analyzed the sentence although it is incorrect. The relations it established between the words are also not correct.

The adjective is a noun that qualifies another noun. In the English language, the adjectives come before the noun being qualified. However, in the Arabic language the adjectives come after the noun being qualified. The adjective in Arabic is termed as "نـعـت" or "صِــــفَــــــةٌ" (property/quality). The noun being qualified is termed as "مـنـعـوتٌ".

The following rules apply to the formation of the Arabic qualifier adjective:

- The adjective "الـنـعـت" follows in gender (feminine / masculine), in state (defined, indefinite) and in number (singular, dual, and plural) the qualified "الـمـنـعـوت" to which it refers.
- The adjective follows the qualified in its case endings.

If we want to apply the rules of the adjective, we must first correct the error when the parser starts to annotate the tokens, certainly the correct sentence is:

"دخل المعلمون المجتهدون"

Such as:

المجتهدون /DTJJS

It should be noted that the DTJJS annotation does not exist in Stanford Parser, where DTJJS is an adjective, and plural with the determiner "Al_ال".

The previous error is called an error of grammatical case endings, which typically appears on the last letter of the stem that indicates its syntactic role.

TABLE. I. SYNTAX ANNOTATION OF THE SENTENCE: " دخل المعلمون المجتهدين"

DOBJ	Direct Object
DEP	Unclassified dependent
VBD	Verb, past tense
DTNNS	noun, plural with the determiner "Al" (ال)
DTJJ	adjective with the determiner "Al" (ال)

Despite the improvements made on parsers, there remains the correction of syntactic errors among its main objectives, also for all NLP systems, and especially for the Arabic language, which is the language for which harmful phrase trees are often syntactically specified. The automatic processing of the error is hindered at several levels [1] because of the complexity, the richness of this language, the absence of short vowel (حركات) in most texts, the irregularity of the word order in the structure of sentences, the agglutination, and the problems of morphological parsing.

Many studies focus on natural language processing at various levels. Such as the morphology [2, 3], the syntax [4, 5] and the semantics [6]. Many programs (as, for example, machine translation, question answering, information retrieval, text summarization, etc.) can exploit them. Nevertheless, the necessity of these programs manifests in the relationship between the words that can be even ungrammatical and can, therefore, lead to incorrect results. This absolutely lacks an effective correction system.

The purpose of our work is to use the results of the Stanford Parser and try to improve them in order to correct syntactic case ending errors based on a logical description of the grammatical relations in the ontology.

The rest of this article is structured as follows. We begin with a brief presentation of the state of the art on parsing and related work in Section 2. In Section 3, we define and describe the grammatical case endings in Arabic language. Section 4 explains the syntactic approach that we adopted as well as the chosen parser. Then, in Section 5, we present in detail our methodology; next, we illustrate our work by an example in Section 6. Section 7 presents the evaluation of our approach. Finally, in Section 8 we draw our conclusion.

II. A BRIEF OVERVIEW ON PARSERS AND RELATED WORK

Parsing has been the subject of several research studies in NLP, so the detection of grammatical errors usually requires parsing the sentence. Checking grammar techniques are classified into three main categories: syntax-based, statistics-based and rule-based analysis.

A. The Syntax-based Approach

This approach usually requires a complete grammar, as well as morphological analysis and syntactic parser. It represents the structure of a sentence according to a universal syntactical scheme. This approach can be classified as a deep syntactical analysis. Among the most famous software that uses it, we find Bokmål (Johannessen et al., 2002) [7]. This is a system used in the Microsoft Word Office XP package released in 2001. The disadvantage of this approach is that it only recognizes if the sentence is incorrect, it will not be able to tell the user where the error is. Thus, if a rule or a constraint of several rules is incorrectly used, the identification of errors will be difficult to apply considering the complexity of natural languages.

B. The Statistics-based Approach

The existence of a large amount of text (corpus) motivated researchers to innovate statistical models to extract linguistic knowledge [8]. Among the linguistic-statistical tools, we find Part Of Speech Taggers (POS) and statistical analyzers. Some parsers use statistical tools to implement various tasks to detect grammar errors, this approach can be used with rules-based techniques such as the Granska parser [9].

These parsers need to be developed in a tagged text to infer a grammar that matches the sentence structure. However, they usually do not lead to reusable grammars and are not always easily interfaced with high-level processing. In addition, the cost of annotation of the learning corpus is far from negligible.

C. The Rule-based Approach

In this latter approach, there is a set of rules developed manually and which corresponds to a text that has been labeled by POS, GramCheck [10] is among the software of this approach. It is similar to the statistics-based approach.

For example, after a determinant, the next word is a noun in 50% of cases in English. This is a kind of interesting rule to add in a system that can predict this situation. Other cases can be solved in a more complex way to incorporate these rules.

However, many special cases are not managed and researchers quickly realized that it was not possible to manage the situation. It is very difficult to capture from simple rules all the complexity of natural languages and all the rules are hand-written, which made the development of the constraint grammar a time-consuming, especially in Arabic language.

D. Related Work

Most Arabic error-correction research is devoted to spelling errors, without worrying about syntactic error correction issues. Indeed, the number of searches (that we know so far) on the correction of syntactic errors in Arabic is very limited.

The Arabic GramCheck [11] is implemented in Prolog (SICStus) for some common grammatical errors. The initial purpose of this tool is to detect the errors and possibly to offer suggestions for improvement. The system is based on deep syntactic analysis and relies on a feature relaxation approach for the detection of ill-formed Arabic sentences. However, there has been no new version since its appearance in 2005.

Recently, a proposed project under-development of a Web-based tool [12] can detect Arabic grammar errors using the Deep Learning, Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) and bidirectional LSTM. Each deep learning model is formed of a set of training data, which will contain labelled sentences. However, the central element of this project is the Arabic corpus, which should not only be annotated for linguistic errors but also indicate types of errors.

Unfortunately, the available corpus does not identify the types of errors, which is a major challenge for this project.

We have also contributed to this field by a new approach based on the automatic generation of correct sentences [13]. This is a very interesting study because we are devoting an important area of Arabic grammar, namely the grammatical case endings.

III. Grammatical Case Endings in Arabic Language

The Arabic language is written from right to left. Its words are generally classified into three principal categories: noun, verb, and particle.

There are two kinds of Arabic sentences [14], nominal and verbal sentence:

- The nominal sentence, where the sentence's first word is a noun (e.g. “الرجل مغربي _ al-rajol maghribi” _ the man is Moroccan).
- The verbal sentence, where the sentence's first word is a verb (e.g. “ولد الرجل في المغرب _ wulida al-rajol fi al-maghrib” _ the man was born in Morocco).

The grammatical case endings “الاعراب” is the change that occurs in the ending of words because of the various factors involved in its grammatical functions.

The Arabic language has three cases endings: the nominative case “المرفوع”, the accusative case “المنصوب” and the genitive case “المجرور” [15].

The following is all the possible situations that indicates when to use these categories:

A. *The Nominative Case*

Nominative case is used in several situations:

- The subject of a verbal sentence.
- The subject and predicate of a nominal sentence.
- The vocative (addressing someone directly).
- The nominative case is also the default for words that are on their own.

B. *The Accusative Case*

Accusative case is used in the following cases:

- The object of a transitive verb.
- Adverbial expressions of time, place, and manner, indicating the circumstances under which an action takes place.
- The internal object or cognate accusative structure. What does that mean? It is just a way of intensifying an action by following the verb with its corresponding verbal noun and an adjective modifying it.
- The circumstantial accusative. This is a way to describe a condition/action going on at the same time as the main action.
- Shows the purpose of an action.
- The accusative of specification.
- The predicate of “*kāna* كان” ‘be’ and its sisters (there are 13 of these light verbs).

C. *The Genitive Case*

We can use the genitive case in these cases

- The object of a preposition
- the object of a locative adverb
- the second term of a genitive construct “إضافة”

Grammatical case endings cause the noun to change in one of the following three ways:

- The final letter's short vowel “حركة” is changed i.e. “مدرسٍ”, “مدرسٌ”, “المدرسُ” or “المدرسِ”.
- A whole letter at the end of the noun is changed i.e. “المدرسون” or “المدرسين”
- Sometime a different form of the noun is used; it is rarely used and only applies to pronouns i.e. “كَ” or “أنتَ”.

In the Arabic language, consonants are always written and short vowels are optional. As a result, written Arabic can be fully, partially, or entirely vowelized. In general, Arabic texts are unvowelized except religious texts, texts used in the education of children and poems. In modern Arabic, some vowels are indicated to help readers remove the ambiguity of certain words.

For this reason, we have agreed to treat unvowelized texts and to deal the apparent errors. At a more general level, errors that can be related to grammatical case endings, we quote:

1) *The sound masculine plurals* “جمع المذكر السالم”

- The final letter is “ن” i.e. “مدرسون” "teachers (male)".
- The second last letter: When the word is nominative, it is the letter “و” i.e. “مدرسون”. When it is accusative or genitive then it becomes the letter “ي” i.e. “مدرسين”.

2) *The dual nouns*

- The final letter in the dual is the letter “ن” i.e. “رجلان” "two men".
- The second last letter: When the word is nominative, it is the letter “ا” i.e. “مدرسان”. When it is accusative or genitive then it becomes the letter “ي” i.e. “مدرسين”.

3) *The six nouns*

- In Arabic are six singular masculine nouns that take various forms: “أَخٌ” brother, “أَبٌ” father, “حَمٌ” father-in-law, “فَمٌ” mouth, “هَنٌ” thing, “ذُوْ” possessor of.

IV. The Syntactic Approach Adopted and the Chosen Parser

There are different formalisms to create the parsing of a text. Besides, nearly all of the research deals with two syntactic representation structures (i.e. constituency and dependency structure).

Parsing by constituents “studied by linguists like Noam Chomsky. (Covington, 2001)” [16], First divide the sentence into several groups of words called syntagmas. A syntagma is an intermediary between the global set, which is the sentence, and the unitary division composed of the words. The principle of this concept is to be able to subdivide the sentence logically into smaller and smaller groups. The syntagma is actually a set of words, or smallest phrases, having a common linguistic role in the sentence (Fig. 1 represents an example).

On the other hand, as we can see, it is difficult to predict the number of phrases that will represent the sentence. It is even possible that different parsers produce syntagmatic representations of different sizes.

In dependency parsing, each word is syntactically dependent on another word in the sentence, except the main verb of the sentence that will be designated as the root of the sentence. Each word, except for the root, has exactly one and one head of which it is dependent. The tree structure can still be used to visually represent dependency parsing.

The words in the sentence represent the nodes of the tree and the dependency relationships represent the arcs connecting these nodes. In Fig. 2, we visually represent the dependency structure of the same sentence shown above, “The student writes the lesson”.

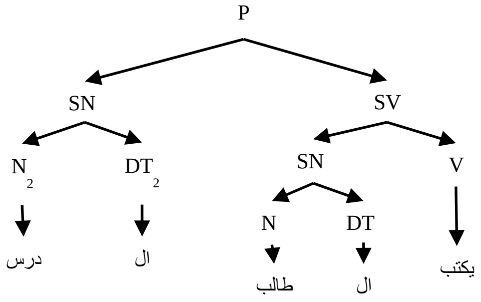


Fig. 1. Syntax of the Sentence: يكتب الطالب الدرس “ the Student Writes the Lesson” Represented by Constituents.

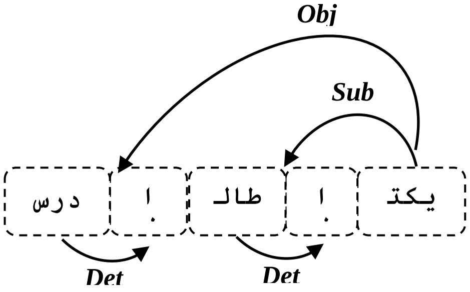


Fig. 2. Syntax of the Sentence: يكتب الطالب الدرس “ the Student Writes the Lesson” Represented by Dependency.

We have noted that dependency parsing provides easy processing, promoting supervised machine learning and the use of classical algorithms [17]. Certainly, dependency trees draw a hierarchical design of structuring information wherever each word is associated with a headword of which it is dependent.

As opposed to constituency parsing, where the number of syntagmas defining the sentence cannot be predicted in advance, each generated parsing contains a fixed number of tree elements. Consequently, recognizing that all words have a single head, the dependency parsing includes precisely one component of representation for each word.

Moreover, since parsing of the Arabic language is often followed by semantic analysis, we seek to facilitate it with more appropriate representation. Now the representation of dependency links is much closer to the semantic links than the syntagmatic representation, this is necessary for the Arabic grammar, indeed there is a strong relationship between the syntactic and semantic aspect of the Arabic language.

Since our work is based on a logical description of the syntactic rules of the Arabic language, our choice fell on the parser: Arabic Stanford Parser [4] in order to set up our methodology. It is based on stochastic non-contextual grammars to solve parsing. It was trained on the Penn Arabic Treebank.

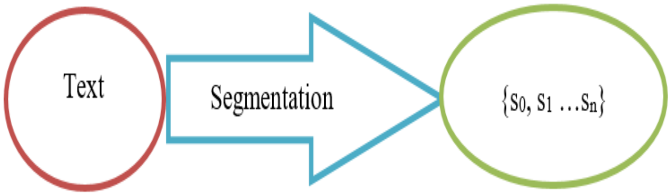
V. Methodology

The syntax correction method adopted is based on the annotation of the Stanford Parser and its syntactical relationships. We can divide this method in 4 steps, which we will detail later:

- Segmentation of text into sentences
- The processing of sentences with the Stanford Parser to obtain the annotation of each word with the syntactic relations between the extracted words.
- The development of correct linguistic relations.
- The detecting and correcting step of grammatical case endings errors.

A. The Segmentation of the Text into Sentences

The segmentation problem of the text into sentences is difficult in the Arabic language. Certainly, it uses neither capital letters nor conventional punctuation that makes the traditional techniques of segmentation adopted in other languages not appropriate to this language. Besides, the agglutination of words is another characteristic of the Arabic that gives segmentation even more challenging to realise (Fig. 3 summarizes this step).



Such as: s_0, s_1, ... are Sentences

Fig. 3. The Segmentation of the Text into Sentences.

B. The Parsing Step

The goal of parsing is summarized in two tasks: creating usable syntactic annotations for the next step the enrichment of linguistic resources for use with our ontology that contains the rules of Arabic syntax.

At this point, we insert the sentences into the Stanford Parser and we draw the label for all language components of the sentence.

The order of linguistic elements

This parser arranges the annotated words using the two fields “index” and “leafNumber” (see Fig. 4), which allows us to know the order of each linguistic element in the sentence, its role appears when there is an ambiguity in the relationships identified by Stanford Parsrer.

```
<S index="0">
    <VP index="1">
         <VBD index="2" value="دخل"
leafNumber="1">
         <NP index="3">
              <DTNNS index="4"
value="المعلمون" leafNumber="2">
              <DTJJ index="5"
value="المجتهدون" leafNumber="3">
         </NP>
    </VP>
</S>
```

Fig. 4. The Order of Linguistic Elements.

C. The Development of Correct Linguistic Relations

This step consists of developing the correct linguistic relations by applying the results of the previous step in order to correct the bad relations by using the Arabic ontology [21], which contains a logical description on the set of rules of the Arabic language (we are going to present later the conception of our ontology).

We adopted the syntactic approach of dependency grammar (DG) founded by Lucien Tesnière. Based on the logic of the predicate we will propose a descriptive formalism of traditional Arabic grammar that will finally take a domain ontology [18].

Grammatical relations fall into two categories:

1) A category for grammatical case: This category shows casuals relations between grammatical components such as subject, object, etc. (Fig. 5 shows an example).

Example:

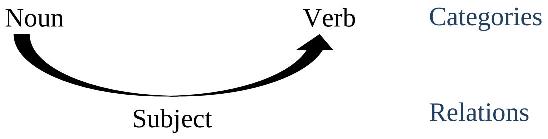


Fig. 5. Categories and Relationships.

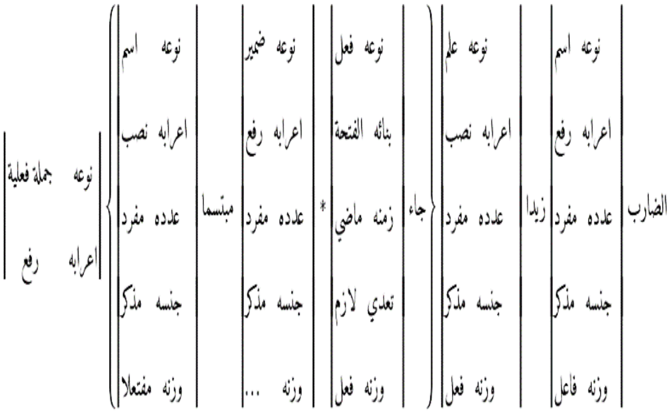


Fig. 6. An Example of a Matrix of Values.

2) A category that represents the characteristics of words: The characteristics of the time, the morphology and the genre of the words such as the relation of gender, which assigns to the words the values of masculinization, feminization, etc. This category will be designed using a matrix of values (see an example in Fig. 6).

D. The Detecting and Correcting Step of Grammatical Case Indings Errors

This step consists of comparing the correct sentence and the original sentence, by calculating a distance of “Levenshtein” [19].

Levenshtein distance:

The Levenshtein distance measures the degree of similarity between two strings (in this case words). It is equal to the minimum number of characters that must be deleted, inserted or replaced to move one string into the other. It is (in the mathematical sense of the term) a positive or zero number.

In our case, we have two possibilities:

- If the system has found that, the original sentence and the corrected sentence with our system are the same, then it moves to the next sentence because it considers that the sentence is correct.
- If the two sentences are not the same, then the system suggests the correct sentence to the user.

VI. Example

The purpose of this section is to illustrate the link that must be made between parsing and correcting detected errors.

We shall consider the following incorrect sentence example:

“جاء المعلمين المدعوين” _ the invited teachers came.

To explain the working of the system as a whole, we will apply our approach in order to correct the errors of grammatical case endings in this sentence:

A. The Segmentation of the Text into Sentences

In this example, we only have one sentence, so the text segmentation is equal to the sentence itself.

The result obtained is:

Segmentation (جاء المعلمين المدعوين) = Seg1 (جاء المعلمين المدعوين)

B. The Parsing Step

After the segmentation, we move to this step, which aims to obtain the annotation of each word as well as the syntactic relations:

جاء/VBD

المعلمين/DTNNS

المدعوين/DTJJ

Root (Root, جاء)

dobj (جاء , المعلمين)

dep (المعلمين , المدعوين)

(S (VP (VBD جاء) (NP (DTNNS المعلمين) (DTJJ المدعوين))))

C. The Development of Correct Linguistic Relations

After the parsing, we have to correct the linguistic elements of these relations:

- The first relation dobj(جاء , المعلمين) is a subject relation that exists in our ontology whose relational space must be a nominative noun (here plural) and the extent of this relation must be a verb, however "المعلمين" is a noun accusative plural noun, with the ending "ين" then we have to change it by "ون" and this piece becomes: "المعلمون"
- The second relation dep(المعلمين , المدعوين) is an adjective relation between a qualifying "المدعوين" and a qualified "المعلمين". As we explained in the introduction, in our ontology qualifying it follows the qualified in its case endings, since we have changed in the first relation the ending of "المعلمون" we must change it also for "المدعوين", then it becomes "المدعوون".

We obtain a result of this step a new correct sentence using our ontology: "جاء المعلمون المدعوون".

D. The Detection and Correction of Case Endings Errors

Finally, we compare the obtained sentence and the original sentence using Levenshtein's distance, which detects the errors in an indirect way, so he proposes the correct sentence.

VII. Evaluation and Discussions

In this section, we present the results of the evaluations performed on Arabic sentences.

There are two main approaches useful to NLP evaluation: black-box and glass-box [20]. In black-box evaluation, the test data are chosen only according to the specified relations between input and output, without considering the internal structure of the tested system. In glass-box evaluation, the evaluator has access to the various workings of the system.

In our work, we have chosen the black-box evaluation approach due to it keeps its original meaning, i.e. evaluation of a system's result "correct sentences" for a given input "incorrect sentences". In this context, the evaluation may not be able to locate the source of the error since it detects them in an indirect way, but it will provide an indication of the good or the bad functioning of the system.

In order to validate our approach, we need to evaluate a set of 100 Arabic sentences, which regroups several types of syntactic endings errors, namely:

- 5 errors of the six nouns.
- 30 errors of disagreement between the circumstantial accusative and the subject.
- 35 errors of disagreement between the adjective and the noun.
- 30 errors of disagreement between the permutative and the antecedent "المبدل و المبدل عنه".

For the reader to understand them easily, and so that linguists will find them simple to evaluate, the majority of these sentences were short and simple.

The average sentence length was three words and the longest sentence was 10 words long. Our system includes approximately 200 grammar rules.

A summary of the evaluation results is presented in Tables II and III. The first column shows the various case endings errors in input sentences.

TABLE. II. Results of Detection and Correction of Case Endings Errors of this Work

Case endings errors	**Detected**	**No detected**	**Total**
Disagreement between the adjective and the noun	34	1	35
Disagreement between the circumstantial accusative and the subject	28	2	30
Disagreement between the permutative and the antecedent	26	4	30
The six nouns	5	0	5
Total (in percentage)	94.28%	5.72%	100

TABLE. III. RESULTS OF DETECTION AND CORRECTION OF CASE ENDINGS ERRORS FROM ANOTHER WORK OF THE SAME RESEARCH TEAM

Case endings errors	**Detected**	**No detected**	**Total**
Disagreement between the adjective and the noun	32	3	35
Disagreement between the circumstantial accusative and the subject	23	7	30
Disagreement between the permutative and the antecedent	26	4	30
The six nouns	4	1	5
Total (in percentage)	85%	15%	100

These results are shown in columns 2 and 3. The output is considered correct if the system gives a syntactically correct sentence.

The results of our approach given in the bottom row show a total detection of 94.28% compared with 85% of the available corpus from another work of the same research team, which is a good level for this type of task. Particularly noteworthy is the high level of precision, which characterizes a very significant level of reliability.

This property of detection is important in this case as correcting sentences because if the system finds that the sentence contains an error, then it will automatically move to the next step to generate the correct sentence.

It would be interesting to compare this work with others. However, there are no available systems on the correction of the grammatical case ending errors in Arabic except for the research study [13] of the same team that developed the present work. Furthermore, it is not practicable to make a comparison for other languages.

To conclude, the correction result is overall very positive. It allows correcting the average errors with 94.28% accuracy.

VIII. CONCLUSION AND PERSPECTIVES

We have presented in this article, a methodology of the detection and automatic correction of syntactic errors and precisely the errors of the case endings in Arabic texts. This work is based on Stanford Parser; the processing of his results is done with the rules and constraints obtained through a logical description of Arabic grammar by an ontology.

It is hoped that the results presented will be useful for the development of Arabic syntax checkers for all errors. We wish also to propose an improvement to Stanford Parser in order to analyze and possibly correct syntactic errors.

ACKNOWLEDGMENT

I would like to express the very thanks to my PhD supervisor, Professor Abderrahim TRAGHA from Hassan II University, who gave me the opportunity to do such research, as well as all the members of the laboratory of Information Technologies and Modelling (LTIM).

REFERENCES

[1] A. Farghaly, K. Shaalan, “Arabic natural language processing: challenges and solutions”, ACM Transaction on Asian Language Information Processing, 8(4), pp.1–22, 2009.

[2] M. Boudchiche, A. Mazroui, Ould Abdallahi Ould Bebah, M.Lakhouaja, A. Boudlal AlKhalil Morpho Sys 2: a robust Arabic morpho-syntactic analyzer J. King Saud Univ. – Comput. Inf. Sci., 29, pp. 141-146, 10.1016/j.jksuci.2016.05.002, 2017.

[3] Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania.LDC Catalog No.:LDC2002L49, 2002.

[4] Chen and Manning, Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. pages 740–750. Association for Computational Linguistics, 2014.

[5] Bikel, D. Intricacies of Collins‟ Parsing Model. Computational Linguistics, 30:4 (479-511), 2004.

[6] C. B. O. Zribi, M. B. Ahmed, "Detection of semantic errors in Arabic texts", Artif. Intell., vol. 195, pp. 249-264, 2013.

[7] J. Johannessen, K. Hagen, P. Lane, “The performance of a grammar checker with deviant language input”, Proceedings of the International Conference on Computational Linguistics (COLING), Taiwan, 2002.

[8] E. Atwell, S. Elliott, “Dealing with Ill-formed English Text in the Computational Analysis of English”, Longman, 1987.

[9] J. Birn, “Detecting grammar errors with Lingsoft’s Swedish grammar checker”, The 12th Nordic Conference in Computational Linguistics, NODALIDA’99, Nordgard T (ed.), Department of Linguistics, Norwegian University of Science and Technology, Trondheim, 2000.

[10] F. Bustamante, F. Leon, “GramCheck: A grammar and style checker”, Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen, 5–9, pp. 175–181, 1996.

[11] K. F. Shaalan, “Arabic GramCheck: a grammar checker for Arabic,” Softw. Pract. Exp., vol. 35, no. 7, pp. 643–665, 2005.

[12] Nora Madi et al. A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. ACLING, 352–355 DOI: 10.1016/j.procs.2018.10.482, 2018.

[13] C. Moukrim, A. Tragha, H. Benlahmer, T. Almalki, “An Innovative Approach to Autocorrecting Grammatical Errors in Arabic Texts”, Journal of King Saud University–Computer and Information Sciences, doi.org/10.1016/j.jksuci.2019.02.005, 2019.

[14] K. Ryding, “Reference Grammar of Modern Standard Arabic”, Cambridge University Press, Cambridge, UK, 2005.

[15] H. Al-Muhtaseb, C. Mellish, “Some Differences Between Arabic and English: A Step Towards an Arabic Upper Model”, The 6th International Conference on Multilingual Computing, Cambridge, UK, 1998.

[16] M. Covington, “A fundamental algorithm for dependency parsing”, Proceedings of the 39th annual ACM southeast conference. Citeseer, pp. 95–102, 2001.

[17] S. Kubler, R. Mcdonald and J. Nivre, “Dependency parsing. Synthesis Lectures on Human Language Technologies”, vol. 1, pp. 1–127, 2009.

[18] D. Klein and D. Christopher, “Fast exact inference with a factored model for natural language parsing”, in Suzanna Becker, vol. 15, pp. 3–10. MIT Press, 2003.

[19] G. V. Bard, “Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric”, in Proceedings of the fifth Australasian symposium on ACSW frontiers, vol. 68, pp. 117-124, Darlinghurst Australia, 2007.

[20] V. Illingworth, “Dictionary of computing”, London, Oxford University Press, 1990.

[21] T. Almalki, “ أنطولوجيا النحو العربي نحو توصيف منطقي
لساني للنحو العربي القديم”, دار النابغة للنشر
والتوزيع, Tanta. Egypt, 2015. http://arabicontology.org/.