

Research Article

An Arabic Grammar Auditor Based on Dependency Grammar

Ameerah Alothman  and AbdulMalik Alsalman 

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Correspondence should be addressed to Ameerah Alothman; a.alothm1n@gmail.com

Received 10 June 2020; Revised 30 September 2020; Accepted 30 October 2020; Published 7 December 2020

Academic Editor: Francesco Bellotti

Copyright © 2020 Ameerah Alothman and AbdulMalik Alsalman. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Arabic language has many complex grammar rules that may seem complicated to the average user or learner. Automatic grammar checking systems can improve the quality of the text, reduce the costs of the proofreading process, and play a role in grammar teaching. This paper presents an initiative toward developing a novel and comprehensive Arabic auditor that can address vowelized texts. We called the “Arabic Grammar Detector” (AGD-أجـد). AGD was successfully implemented based on a dependency grammar and decision tree classifier model. Its purpose is to extract patterns of grammatical rules from a projective dependency graph in order to designate the appropriate syntax dependencies of a sentence. The current implementation covers almost all regular Arabic grammar rules for nonvowelized texts as well as partially or fully vowelized texts. AGD was evaluated using the Tashkeela corpus. It can detect more than 94% of grammatical errors and hint at their causes and possible corrections.

1. Introduction

As with all languages, researchers into the Arabic language and its applications have exerted strenuous efforts to achieve progress in language processing. These efforts have focused on multiple levels, including morphology [1, 2], which studies and characterizes the structure of words, syntax [3, 4], the grammatical arrangement of words, and semantics [5, 6], which determines a text’s exact meaning. Question answering, grammar detection and correction, machine translation, and automatic summarization are types of applications dependent on these level processes. Some of these applications are extremely important and form the basis for many natural language processing (NLP) applications, such as spelling and grammar auditing systems.

There is a lack of research in the field of Arabic grammar detection and correction. This is due to the specific characteristics of the Arabic language, such as its complexity and richness at the semantic level, which can lead to ambiguity and produce an erroneous and incomprehensible text. Moreover, the flexible arrangement of words in a sentence, the properties of agglutination, and diacritics complicated Arabic grammar. All these properties lead to a variety of issues on the morphological, syntax, and semantic levels.

Moreover, one important reason for the lack of research in this area is the incomplete infrastructure for the Arabic morphological level. This has led to a lack of breakthroughs in research at any higher level, namely, syntax and semantic. The emphasis is still placed on studies related to the morphological level, such as spell-checking and correct word analysis.

We attempt here to bridge the research gap by implementing a comprehensive supervised learning system for detecting grammatical errors and hint at their causes and correction in diacritic and nondiacritic Arabic text. The dependency grammar of the Arabic grammatical rules adopted in this research enables us to parse the Arabic structure graph to infer the correct pattern of grammatical rules for a sentence based on the properties of its words, which are extracted via a morphological analyzer. As far as supervised learning is concerned, part of the task of the decision tree learning algorithm involves learning how to classify the words of a text according to its grammar parsing status (correct-incorrect). This work is considered the first initiative to automate the dependency grammar rules of the Arabic language. As far as we know, AGD is the first Arabic grammatical auditing system dealing with the diacritic Arabic text.

The rest of the paper is structured as follows. Section 2 presents the literature review, Section 3 demonstrates dependency grammar, and Section 4 describes grammatical rules adopted in our system. Section 5 features a demonstration of the syntax dependency and classification model used. Section 6 presents an evaluation of the proposed system. Finally, Section 7 concludes the paper with some future directions.

2. Literature Review

There are numerous specialized research studies for auditing language-specific grammar, such as English [7–16], European languages [17–23], and Asian languages [24–29]. Moreover, there are many well-known grammar checking tools, such as Grammarly [30], LanguageTool [31], and ClearEdits [32]. Yet, none of them supports the Arabic language, which is one of the six official languages of the United Nations. To the best of our knowledge, there is no available Arabic grammar checker so far [33].

From a research point of view, only three studies, to our knowledge, have dealt with grammar auditing in Arabic. The first was conducted by Shaalan [34], who worked on the first Arabic grammar checker, called Arabic GramCheck. The study was divided into two parts: a morphological analyzer and a standard bottom-up parser. The author evaluated this system on personal data. Unfortunately, this checker is no longer available.

The second study on Arabic grammar checkers was presented by Moukrim et al. [35]. Apparently, their system uses the Arabic grammar described in the ontology [36] to generate constraints and sentence rules. Their work is based on two hypotheses. First, from the ontology, all possible sentences can be generated, and second, all correct generated sentences can be compared with the original sentence. Some improvements have been made to this work, as demonstrated in [37], using a Stanford parser to segment and annotate the input sentences. Then, they adopted the same two assumptions they mentioned in [35] to detect grammatical errors. The researchers ignored most cases of grammatical errors in the Arabic language and evaluated only four simple types of errors on nonvowelized texts. These errors include the end-mark of the five nouns (الأسماء الخمسة), the syntactic dependency in the adjective, the adverb, and the permutation with its noun. The grammar check was 94.28% accurate on the four types of errors, using only 100 sentences.

Lastly, a recent preliminary work by Madi and Al-Khalifa [38] examines the use of deep learning to detect grammatical errors in Arabic. They used the corpus of the Qatar Arabic Language Bank (QALB) for their training and assessment stages. The results of this study have not been published yet.

3. Dependency Grammar (DG)

One of the tasks of language studies is to identify the proper grammatical syntax for every sentence within a specific formalism and grammar [39]. Thus, formalism implies the correct structural constraints of the language. However, the

grammar of the language is made up of two parts. First is the constituency grammar (النحو والمكونات), which directly parses the sentence in terms of its components with context-free grammar (CFG). Second is the dependency grammar (النحو والعلاقات), which treats the sentence according to the syntactic relationships between words [36].

A DG typically falls into the class of grammars that focus on words rather than constituents. Grammars that are mainly built on constituents are known as phrase structure grammars. Thus, phrase structure grammars are constituent-based, whereas DGs are word-based. Unlike the phrase structure grammars, which see sentences and clauses structured in terms of constituents, DGs presume that the structure of sentences and clauses results from the dependency between words [40, 41]. Graphically, the dependency grammar describes the structure of sentences as a tree where nodes represent words and edges represent dependencies. In contrast, the terminals and the non-terminals of context-free grammar in the constituency grammar are represented by the leaf's nodes and internal nodes, respectively, as shown in Figure 1.

Natural language dependency representations are flexible and simple; they use directed graphics to encode words and their syntax dependencies. Figure 2 illustrates an example of a dependency graph for the sentence “The boy reads the book in the morning”—الولد الكتّاب صباحاً يقرأ. The edges of this graph represent a unique syntax dependency pointing from a word to its modifier. In this depiction, all edges are tagged with the specific syntax function of the dependency—for example, sub for the subject and obj for the object of a verb. In order to make programming calculations and some important definitions easier, a dummy token is inserted into the sentence as the rightmost word. It will, therefore, always be the root of the dependency graph.

The dependency graph is an example of a nested or projective graph. Assuming that the root of the graph is the rightmost word in the sentence (Arabic is written from right to left), a projective graph is a graph whose edges can be drawn in the plane above the sentence without two edges crossing each other, as shown in Figure 2. Meanwhile, a nonprojective dependency graph cannot fulfil this property [42]. An example of a nonprojective graph for the sentence “The boy is reading the book in the morning”—الولد يقرأ الكتّاب صباحاً—is shown in Figure 3. Long-distance dependencies, or dependencies in languages with flexible word order, are the reason for nonprojectivity. An important part of sentences in some languages necessitates a nonprojective dependency analysis. Therefore, the ability to learn and derive nonprojective dependency graphs is an essential step in solving certain problems—for example, multilingual processing of languages or permutation cases (حالات التباديل) in the Arabic language.

Some studies [43, 44] assumed another type of grammar called a link grammar as a third type of grammatical formalisms. As far as we know, the main difference between the link grammar and the dependency grammar is that the former must have an algorithm to demonstrate how to link

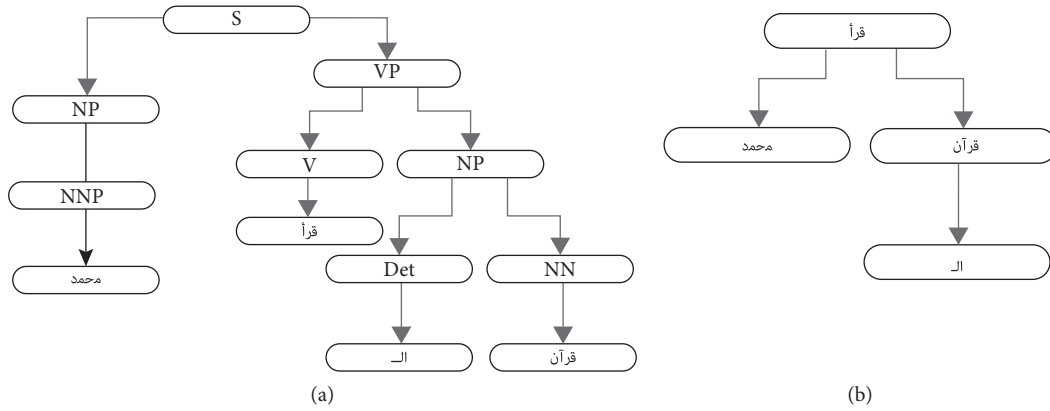


FIGURE 1: Constituency and dependency structures for the Arabic phrase **قرأ محمد القرآن**.

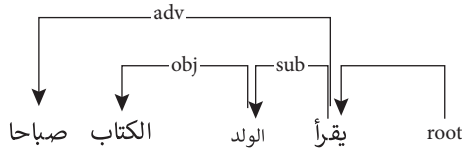


FIGURE 2: An illustration of a projective dependency graph.

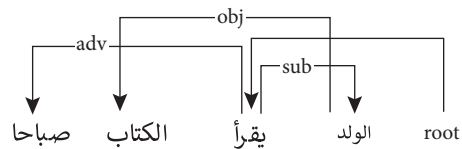


FIGURE 3: An illustration of a nonprojective dependency graph.

words, while the latter adopts the dependency principle to link two or more words.

4. Grammatical Rules

This section introduces grammatical rules and how to output their proper end-mark by our system. To the best of our knowledge, there is no work to automate the Arabic grammatical rules, except the Arabic grammar ontology built by [36] [المالك], which has many restrictions. Thus, our work seeks to construct the Arabic grammar rules as a dependency grammar, based on the criteria of grammatical rules defined in *متن الأجروميّة* and its explanations by [45] [العثيمين], which is considered as a basic reference for Arabic grammar. These grammatical rules are laid out in a hierarchy, grounded in the most generic and dependent properties. Each rule has its own properties and constraints. Once these properties and constraints between a head (current word) and its dependencies (previous words) can be achieved completely, rules lower in the hierarchy can be used. For example, the head may be a noun, verb, or particle. The noun is then divided into subject and inchoative rules, and so on. The complete hierarchy of our grammatical rules is shown in Figure 4.

In total, the hierarchy contains 30 main grammatical rules, and some of these rules have one or more subrules. At

any moment, this backbone of the hierarchy can be extended (by adding subrules) to include several extensions and enhancements intended to facilitate and improve the usage in certain applications.

5. Syntax Dependency and Classification Model

In any sentence, the relationship between two or more words (dependencies) with one word (head) is called a syntax dependency. Our technique for generating syntax dependencies is mainly based on patterns of grammatical rules applied to sentence structure graphs with the aim of constructing a decision tree classifier model. Even though the approach is general, it requires relevant rules for each language and treebank representation. The method for verifying the correct grammatical syntax dependencies—in order to check whether the end-mark (العلامّة الإعرابية) is correct—consists of three phases: dependency extraction, dependency synthesis, and predicate the correct end-mark. Figure 5 illustrates these phases.

The dependency extraction phase is straightforward. First, a sentence is analyzed with a morphological analyzer. A morphological analyzer identifies the attributes of a particular word by a chain of the morphological processes that the word has undergone, including specifying the word units and how these units are related to form the words overall, the function of these units in a particular word, and its syntactic and semantic behaviour in the sentence. In our case, any morphological analyzer could be used, but in practice, we have used MADAMIRA [46], which was developed by the Penn Arabic Treebank (PATB). MADAMIRA provides high-quality word-level disambiguation of Arabic expression and presents a high-precision statistical analysis of sentence structure. For more information about the morphological step using MADAMIRA, please see [47]. Second, a root is identified to find the right dependencies in the sentence. In this version of the work, a projective dependency graph was adopted, which means a word and its descendants formed a contiguous substring of the sentence. Therefore, we assume that the root of the dependency graph was the proper word on the far right of the sentence. This is sufficient to represent most of the rules of Arabic grammar,

<p>The head</p> <p><u>Noun</u></p> <p>Subject الفاعل Nominative_end-mark</p> <p>Pro_agent نائب الفاعل Nominative_end-mark</p> <p>Inchoative المبتدأ Nominative_end-mark</p> <p>Predicate الخبر Nominative_end-mark</p> <p>Noun_of 'kana and its sister'* اسم كان وأخواتها Nominative_end-mark</p> <p>Predicate_of 'inna and its sister'** خبر إن وأخواتها Nominative_end-mark</p> <p>Object المفعول Accusative_end-mark</p> <p>Adverb الحال Accusative_end-mark</p> <p>Specification التمييز Accusative_end-mark</p> <p>Exception المستثنى Accusative_end-mark</p> <p>Noun_of la_categorical_negation اسم لا النافية للجنس Accusative_end-mark</p> <p>Vocative المنادى Accusative_end-mark</p> <p>Predicate_of 'kana and its sister'* خبر كان وأخواتها Accusative_end-mark</p> <p>Noun_of 'inna and its sister'** اسم إن وأخواتها Accusative_end-mark</p> <p>Genitive_by_preposition مجرور بحرف الجر Genitive_end-mark</p>	<p>Genitive_by_addition مجرور بالإضافة Genitive_end-mark</p> <p>Adjective النعت Nominative_end-mark Accusative_end-mark Genitive_end-mark</p> <p>Emphasis التوكيد Nominative_end-mark Accusative_end-mark Genitive_end-mark</p> <p>Permutation البدل Nominative_end-mark Accusative_end-mark Genitive_end-mark</p> <p>Conjunction العطف Nominative_end-mark Accusative_end-mark Genitive_end-mark</p> <p><u>Verb</u></p> <p>Nominative_imperfective المضارع المرفوع Nominative_end-mark</p> <p>Accusative_imperfective المضارع المنصوب Accusative_end-mark</p> <p>Jussive_imperfective المضارع المجزوم Jussive_end-mark</p> <p>Conjunction العطف Nominative_end-mark Accusative_end-mark Jussive_end-mark</p> <p><u>Particle</u></p> <p>Active_partical حرف عامل</p> <p>Inactive_partical حرف غير عامل</p>
--	---

* 'kana and its sisters' are a special set of verbs in the Arabic language. If anyone of which precedes a nominal sentence, it makes the inchoative nominative and the predicate accusative. Kana and its sisters are [ما أنفك ، ما فنى، ما زال، ما برح، ما دام، ما ليس، بانت، ظل، أمسى، أضحى، أصبح، صار، كان، صار، أصبح، أضحى، أمسى، ظل، بانت، ليس، مادام، ما برح، ما زال، ما فنى، ما أنفك].

** 'inna and its sister' are a special set of particles in the Arabic language. If anyone of which precedes a nominal sentence, it makes the inchoative accusative and the predicate nominative. Inna and its sisters are [إن ، أن ، كأن ، لكن ، ليت ، لعل ، لا].

FIGURE 4: The grammatical rules hierarchy.

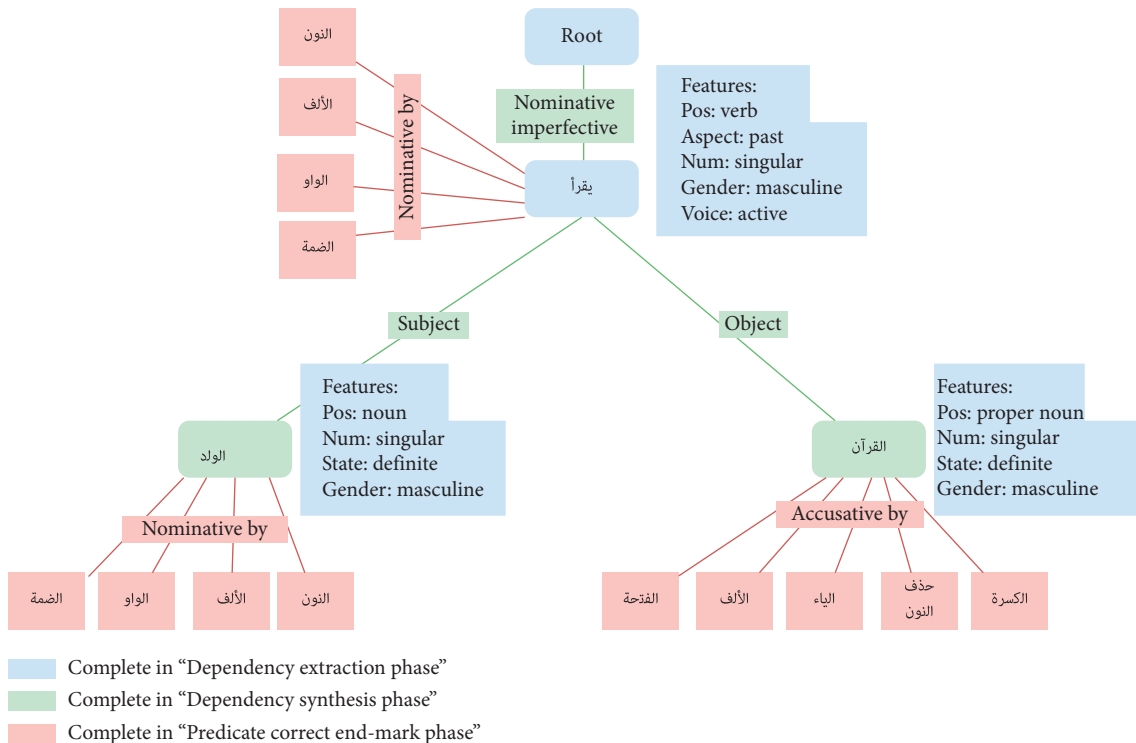


FIGURE 5: Phases of verifying the end-mark correctness based on grammatical syntax dependencies.

except the odd permutation cases. The Arabic language is very flexible in terms of words' ordering. These different word orderings lead to many permutation cases. The position of the root in these cases is changed; it cannot be assumed to be at the far right of the sentence. For example, in Figure 3, the subject (“the boy”—الولد) precedes (occurs right in Arabic sentence) the verb (“read”—قرأ), which is valid syntax in the Arabic language. In this case, if we consider the subject as the root of the sentence, it will lead to wrong results.

After completing this phase, the word-level processing is finished, and the processing will move to sentence-level in the next phases.

In the second phase, dependency synthesis, we associate each of the extracted dependencies with a grammatical rule (listed in the hierarchy in Figure 4), which is accurately designated as possible to infer the right syntax dependency. The dependencies are flexible to accommodate any sentence's syntax in the Arabic language. Once each grammatical rule was extracted, we define one or more projective dependency graphs based on the sentence structure. Conceptually, each node in the graph is tested to conform to the properties and constraints of the extracted grammatical rule, and the graph corresponding to the most specific grammatical rules is considered to be the type—or name—of the syntax dependencies.

In the last phase, the classification model is completed by reaching the leaf's nodes, which represent the end-mark cases to the specific grammatical dependency rule produced in the synthesis phase. Through that, the correct end-mark could be inferred by comparing the end-mark of the input word with the correct end-mark extracted from the classifier model, and errors—if they occur—are presented to the user. It is worth noting that the checker will recognize the grammatical errors and inform the user what the exact cause of errors is. Therefore, extra clarification rules were added to each grammatical rule to adopt this feature.

Thus, if we insert a dummy root for the sentence, then each word (node) in the internal nodes of the decision tree will have a dependency relation. Accordingly, the number of syntax dependencies in the representation equals the number of words in the sentence. In grammar auditors, it is necessary to combine the particle (such as prepositions and conjunctions) with the next word to form one word.

6. Evaluation

In this section, we discuss the results obtained by AGD. The evaluation consists of two stages: a preliminary evaluation and an evaluation of inserted errors.

The evaluations were performed on 10 essays that contain 752 sentences. The essays were selected from different collections of partially or totally vowelized Arabic texts, taken from the Tashkeela corpus (<https://sourceforge.net/projects/tashkeela/>). Table 1 shows the size of each essay, including the type it belongs to.

In the evaluation process, we adopted four symbols in order to facilitate the steps of the performance computation in an accuracy, recall, and precision metric:

TABLE 1: Description of test essays.

Essay type	Essay #	# of sentences
Book texts	S01	51
	S04	180
	S05	101
News texts	S06	72
	S07	71
	S08	60
	S02	49
General texts	S03	78
	S09	48
	S10	42

Correct Results:

TT: the word is correct (True), and the system says it is correct (True).

FF: the word is incorrect (False), and the system says it is incorrect (False).

Incorrect Results:

TF: the word is correct (True), and the system says it is incorrect (False).

FT: the word is incorrect (False), and the system says it is correct (True).

Additionally, during the construction of the AGD, some errors may not be discovered due to incorrect analysis in the preprocessing operation, or due to ambiguity in semantics. These errors are outside the framework of the AGD because they happened in the preprocessing stage that precedes the grammar auditor. We called these errors limitations. Based on that, several limitations were set to enable the AGD to work in an ideal situation, i.e., assuming no errors happened in the preprocessing phases. These limitations are as follows:

- (A) It represents a subject that is a hidden pronoun (الفاعل ضمير مستتر). Usually, the subject comes directly after (occurs left in Arabic sentence) the verb, yet in some cases, the subject is hidden, and the word after the verb is an object. This case can only be detected at the semantic level. For example, “Mohammad read the book and wrote the homework”—قرأ محمد الكتاب، وكتب الواجب; the subject in the first sentence is Mohammad, while in the second one, it is a hidden pronoun.
- (B) It represents an indefinite added to indefinite (المضاف إلى نكرة). The structures of the Arabic language allow an indefinite word to be added to a definite or an indefinite word. AGD can distinguish the addition to the definite word (e.g., “the grammar book”—كتاب النحو), but not to an indefinite word (e.g., “a grammar book”—كتاب نحو).
- (C) It represents errors in morphological analysis. Morphological analysis errors are common, and they affect the correctness of the grammatical structure—for example, not analyzing some words, or analyzing a noun as a verb or vice versa.
- (D) It represents an odd position of the subject; i.e., the object precedes the subject (e.g., “the book read by

Mohammad”—(قرأ الكتاب محمد) or the subject precedes the verb (e.g., “Mohammad read the book”—(محمد قرأ الكتاب).

Table 2 displays the limitations that were discovered in the texts of tested essays. We should mention that determining the syntax dependencies of Arabic grammatical rules correctly depends on accurate morphological analysis and determination of the semantics of the sentence. In AGD, the semantic understanding (limitations A, B, and D) and the incorrect analysis (limitation C) represent 82.86% and 17.14% of total limitations, respectively.

Before presenting the results of our work in detail, some points must be highlighted, namely:

- (1) The evaluated texts were free of spelling errors because they affect the morphological process accuracy, which is a step that precedes the grammar auditing process.
- (2) We cannot compare our work with others because none have used a well-known corpus, and none is available for testing and comparing.
- (3) The Tashkeela corpus contains more than 75 million words. Our evaluation aims to measure the effectiveness of our approach. Accordingly, random essays were selected from the corpus to fulfil that aim. The corpus includes (totally or partially) vowelized text, as well as nonvowelized texts.

6.1. Stage 1: Preliminary Evaluation. This evaluation aims to evaluate AGD in detecting and correcting errors that we do not know before the auditing process. For a comparison purpose, a specialized human auditor was hired to do an audit for the same samples. It is worth noting here that the results of the human auditor and the AGD audit were finally examined by a professor in the field of Arabic grammar to judge on the results if they differ. This step was taken due to the divergence and complexity of the Arabic language. The results of the human audit and AGD audit, along with the number of words that were grammatically audited (GA) correctly in each essay, are shown in Tables 3 and 4.

From Tables 3 and 4, the most grammatical errors appeared when the related dependent was apart; for example, in “The boy wrote the math homework with the teacher in the morning, science and language”—(كتب الولد واجب الرياضيات مع المعلم صباحا والعلوم واللغة). In this sentence, science and language are related to math, and they should take its end-mark. The second most common error is determining the adverb object rather than the general noun; for example, in “December 18, International Day of the Arabic Language”—(ديسمبر اليوم العالمي للغة العربية), the day is a noun, not an adverb object.

The confusion matrices in Tables 5 and 6 show that 43 words are grammatically audited incorrectly by AGD in its ideal situation, compared with 32 words by human audit. Most of the human audit errors resulted from incorrect discrimination of the end-mark, as well as inaccurate identification of the accusative and jussive cases. The

TABLE 2: Limitations in the “Arabic Grammar Detector” (AGD) audit.

Essay #	A	B	C	D
S01	6	3	4	1
S02	11	4	0	0
S03	9	9	8	0
S04	1	3	7	0
S05	1	5	1	0
S06	6	8	1	4
S07	5	10	1	2
S08	9	7	1	2
S09	0	1	1	0
S10	6	3	0	0
Total	54	53	24	9

TABLE 3: Accuracy of the human audit.

Essay#	#Word	Correct		Incorrect		Accuracy (correct/#Words) (%)
		TT	FF	TF	FT	
S01	246	243	2	0	1	99.59
S02	277	268	3	0	6	97.83
S03	622	600	12	5	5	98.39
S04	546	540	0	0	6	98.90
S05	410	405	3	0	2	99.51
S06	495	494	1	0	0	100.00
S07	528	527	0	0	1	99.81
S08	485	478	3	1	3	99.18
S09	227	226	1	0	0	100.00
S10	273	270	1	0	2	99.27
Total	4109	4051	26	6	26	99.22

precision and recall in Table 7 show that the AGD has a high recall but low precision, which means it detects most of the errors out of total errors, but some of its predicted errors are incorrect. In contrast, the human audit detects half of the errors out of the total errors, but most of the predicted errors are correct.

6.2. Stage 2: Evaluation of Inserted Errors. This evaluation aims to evaluate AGD in discovering inserted errors in fully correct essays and being able to explain their causes. We measure the performance of AGD in terms of precision and recall. Admittedly, there is no available corpus for grammatical errors in the Arabic language. Accordingly, we manually inserted many grammatical errors in the evaluated texts (e.g., (يقرأ الولد القرآن) changed to (يقرأ الولد القرآن)). The errors were embedded by persons who were not members of the AGD programming team. The results of the AGD audit, along with the number of inserted errors in each essay, are shown in Table 8.

As shown in Tables 9 and 10, only 13 out of 220 errors were not GA correctly by AGD in its ideal situation. In other words, the failure of AGD to correct the grammar of diacritic Arabic texts by grammar correction is only 5.91%. Most of these errors focused on the precise identification of the adjective-noun following the descriptor or on the sources of

TABLE 4: Accuracy of AGD audit in stage 1.

Essay#	#Word	Correct		Incorrect		Accuracy (correct/#Words) (%)
		TT	FF	TF	FT	
S01	246	243	1	1	1	99.19
S02	277	268	6	3	0	98.92
S03	622	607	5	4	6	98.39
S04	546	540	6	0	0	100.00
S05	410	406	2	2	0	99.51
S06	495	491	0	3	1	99.19
S07	528	518	1	9	0	98.30
S08	485	476	3	4	2	98.76
S09	227	222	0	5	0	97.80
S10	273	269	2	2	0	99.27
Total	4109	4040	26	33	10	98.95

TABLE 5: Confusion matrix for human audit.

	GA correct	Not GA correct
GA	FF = 26	FT = 26
Not GA	TF = 6	TT = 4051

TABLE 6: Confusion matrix for AGD audit in stage 1.

	GA correct	Not GA correct
GA	FF = 42	FT = 10
Not GA	TF = 33	TT = 4024

TABLE 7: Precision and recall for the human and AGD audit in stage 1.

Feature	Precision (FF/(FF + TF)) (%)	Recall (FF/(FF + TF)) (%)
Human audit	81.25	50.00
AGD audit	56.00	80.77

TABLE 8: Accuracy of AGD audit in stage 2.

Essay#	Word#	# Inserted errors	Correct		Incorrect		Accuracy (%)
			TT	FF	TF	FT	
S01	246	13	232	13	1	0	99.59
S02	277	55	219	51	3	4	97.47
S03	622	15	603	14	4	1	99.20
S04	546	15	531	15	0	0	100.00
S05	410	29	379	28	2	1	99.27
S06	495	19	473	17	3	2	98.99
S07	528	20	499	19	9	1	98.11
S08	485	24	457	21	4	3	98.56
S09	227	15	207	14	5	1	97.36
S10	273	15	256	15	2	0	99.27
Total	4109	220	3856	207	33	13	98.88

TABLE 9: Confusion matrix for AGD audit in stage 2.

	GA correct	Not GA correct
GA	FF = 207	FT = 13
Not GA	TF = 33	TT = 3856

TABLE 10: Precision and recall for the human and AGD audits in stage 2.

Feature	Precision (FF/(FF + TF)) (%)	Recall (FF/(FF + TF)) (%)
AGD audit	86.96	94.42



FIGURE 6: The AGD demo.

verbs (الاسم المصدري الذي يعامل معاملة الفعل) that take the place of the verbs and treat the noun after them according to the properties of the verb, not the noun.

7. Conclusions

An Arabic grammar auditor is a complex system that requires extensive linguistic resources and research, as well as the help of specialists in the field. An in-depth grammar auditor that adopts a new approach is presented in this paper. It aims to detect grammatical errors and hint at their causes and correction in both vowelized and nonvowelized Arabic texts. This approach is based on the grammar of the projective dependency graph, whose rules are extracted from the hierarchy of grammatical rules that we have constructed. In brief, a synthesis of the dependencies is performed to designate the most specific pattern of grammatical rules, depending on the sentence words' features, to infer the proper end-mark based on the results of the decision tree classifier model. We have achieved promising results using this approach. In its current stage, AGD provides accurate results compared with human auditing. The appendix gives a sample demonstration of AGD.

The future work of our research includes three aspects. The first aspect is detecting errors related to agreements in the Arabic language, such as the agreement between the verb and the subject's gender and agreement between the inchoative and the predicate in number and gender. This can be done by extending the backbone of the hierarchy of Arabic grammatical rules adopted in our system. The second aspect is expanding the scope of AGD evaluation by

uploading it on the web to be available for public use and benefiting from users' feedback to enhance the performance of the AGD system. The third aspect is enhancing the AGD by involving the nonprojective dependency graph to deal with the Arabic odd permutation cases.

Appendix

Figure 6 shows a screenshot of the AGD demo. It shows the Arabic grammatical audit of the vowelized input text. The red text color code indicates grammar errors. The other tab panels show the morphological analysis results. Hovering over a word (as is done here on **الأطفال**) displays a box with an explanation and correction of the error for that word.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the Deanship of Scientific Research at King Saud University for funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR). The authors also would like to thank Abdullah Alansary, the professor in the field of

Arabic grammar, Imam Mohammad bin Saud University, for directing the search and helping in the audit process.

References

- [1] I. A. Al-Sughayer and I. A. Al-Kharashi, "Arabic morphological analysis techniques: a comprehensive survey," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 3, pp. 189–213, 2004.
- [2] A. Alonthman and A. M. Alsalman, "Arabic morphological analysis techniques: a survey and classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 214–222, 2020.
- [3] S. Green and C. D. Manning, "Better Arabic parsing: baselines, evaluations, and analysis," in *Proceedings of the Coling 2010-23rd International Conference on Computational Linguistics*, vol. 2, pp. 394–402, Beijing, China, August 2010.
- [4] Y. Zaki, H. Hajjar, M. Hajjar, and G. Bernard, "A survey of syntactic parsers of Arabic language," in *Proceedings of the ACM International Conference Proceeding Series*, pp. 1–10, Beijing, China, October 2016.
- [5] A. M. Al-Zoghby, A. S. E. Ahmed, and T. T. Hamza, "Arabic semantic web applications-a survey," *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 1, pp. 52–69, 2013.
- [6] A. M. Al-Zoghby and K. Shaalan, "Semantic search for Arabic," in *Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference, FLAIRS*, vol. 2015, pp. 524–529, Melbourne, FL, USA, May 2015.
- [7] J. C. Park, M. Palmer, and G. Washburn, "An English grammar checker as a writing aid for students of English as a second language," in *Proceedings of the Fifth Conference on Applied Natural Language Processing: Descriptions of System Demonstrations and Videos*, p. 24, Washington, DC, USA, March 1997.
- [8] E. M. Bender, "Arboretum: using a precision grammar for grammar checking in call," in *Proceedings of the Instil/ical Symposium*, Tokia: Venice, Italy, January 2004.
- [9] C. Brockett, W. B. Dolan, and M. Gamon, "Correcting ESL errors using phrasal SMT techniques," in *Proceedings of the COLING/ACL 2006-21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 249–256, Sydney, Australia, July 2006.
- [10] M. Chodorow, J. R. Tetreault, and N.-R. Han, "Detection of grammatical errors involving prepositions," in *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pp. 25–30, Prague, Czech Republic, June 2007.
- [11] A. Kumar and S. B. Nair, "An artificial immune system based approach for English grammar checking," in *Proceedings of the International Conference on Artificial Immune Systems*, pp. 348–357, Santos, Brazil, August 2007.
- [12] B. Crysmann, N. Bertomeu, P. Adolphs, D. Flickinger, and T. Klüwer, "Hybrid processing for grammar and style checking," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 153–160, Manchester, UK, August 2008.
- [13] D. Dahlmeier and H. T. Ng, "Grammatical error correction with alternating structure optimization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 915–923, Portland, OR, USA, June 2011.
- [14] M. Felice, Z. Yuan, Ø. E. Andersen, H. Yannakoudakis, and E. Kochmar, "Grammatical error correction using hybrid systems and type filtering," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, Baltimore, MD, USA, June 2014.
- [15] J. Ji, Q. Wang, K. Toutanova, Y. Gong, S. Truong, and J. Gao, "A nested attention neural hybrid model for grammatical error correction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 753–762, Vancouver, Canada, July 2017.
- [16] Z.-R. Liu and Y. Liu, "Exploiting unlabeled data for neural grammatical error detection," *Journal of Computer Science and Technology*, vol. 32, no. 4, pp. 758–767, 2017.
- [17] T. Holan, V. Kubon, and M. Plátek, "A prototype of a grammar checker for Czech," in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 147–154, Washington, DC, USA, March 1997.
- [18] V. Kovář, *Partial Grammar Checking for Czech Using the SET Parser*, LNAI Springer, vol. 8655, pp. 308–314, Lagodekhi, Georgia, 2014.
- [19] M. Miłkowski, "Developing an open-source, rule-based proofreading tool," *Software: Practice and Experience*, vol. 40, no. 7, pp. 543–566, 2010.
- [20] F. R. Bustamante and F. S. León, "GramCheck: A grammar and style checker," in *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pp. 175–181, Vancouver, Canada, July 1996.
- [21] J. Kinoshita, L. do Nascimento Salvador, and C. E. D. de Menezes, "CoGrOO: A Brazilian-Portuguese grammar checker based on the cetenfolha corpus," in *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 2190–2193, Genoa, Italy, May 2006.
- [22] J. Birn, "Detecting grammar errors with Lingsoft's Swedish grammar checker," in *Proceedings of the 12th Nordic Conference of Computational Linguistics (NODALIDA 1999)*, pp. 28–40, Trondheim, Norway, December 2000.
- [23] P. Gakis, C. Panagiotakopoulos, K. Sgarbas, C. Tsalidis, and V. Verykios, "Design and construction of the Greek grammar checker," *Digital Scholarship in the Humanities*, vol. 32, no. 3, pp. 554–576, 2017.
- [24] A. Fahda and A. Purwarianti, "A statistical and rule-based spelling and grammar checker for Indonesian text," in *Proceedings of the 2017 International Conference on Data and Software Engineering (ICoDSE)*, pp. 1–6, Palembang, Indonesia, November 2017.
- [25] M. P. Go, N. Nocon, and A. Borra, "Gramatika: A grammar checker for the low-resourced Filipino language," in *Proceedings of the TENCON 2017-2017 IEEE Region 10 Conference*, pp. 471–475, Penang, Malaysia, November 2017.
- [26] M. S. Gill and G. S. Lehal, "A grammar checking system for Punjabi," in *Proceedings of the 22nd International Conference on Computational Linguistics: Demonstration Papers*, pp. 149–152, Manchester, UK, August 2008.
- [27] S. K. Sharma and G. S. Lehal, "Improving existing punjabi grammar checker," in *Proceedings of the 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, pp. 445–449, New Delhi, India, March 2016.
- [28] L. Bopche, G. Dhopavkar, and M. Kshirsagar, "Grammar checking system using rule based morphological process for an Indian language," in *Proceedings of the International Conference on Computing and Communication Systems*, pp. 524–531, Washington, DC; US, February 2011.
- [29] N. Ehsan and H. Faili, "Towards grammar checker development for Persian language," in *Proceedings of the 6th*

- International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010)*, pp. 1–8, Beijing, China, August 2010.
- [30] “Grammarly: Free Online Writing Assistant.” [Online]. Available: <https://www.grammarly.com/>. [Accessed: 31-May-2020].
- [31] “LanguageTool - Spell and Grammar Checker.” [Online]. Available: <https://languagetool.org/>. [Accessed: 31-May-2020].
- [32] “ClearEdits writing editing software: go beyond grammar check | grammar check software.” [Online]. Available: <https://www.clearwriter.com/clearedits.html>. [Accessed: 31-May-2020].
- [33] A. Alqahtani and A. Alsaif, “Automatic evaluation for Arabic essays: a rule-based system,” in *Proceedings of the 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 1–7, Ajman, UAE, December 2019.
- [34] K. F. Shaalan, “Arabic GramCheck: a grammar checker for Arabic,” *Software: Practice and Experience*, vol. 35, no. 7, pp. 643–665, 2005.
- [35] C. Moukrim, A. Tragha, E. H. Benlahmer, and T. Almalki, “An innovative approach to autocorrecting grammatical errors in Arabic texts,” *Journal of King Saud University-Computer and Information Sciences*, 2019.
- [36] ط. المالك، أنطولوجيا حاسوبية للنحو العربي، دار النابغة، مصر، 2015.
- [37] C. Moukrim, A. Tragha, E. Habib, and E. Habib, “The correction of the grammatical case endings errors in Arabic language,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, pp. 279–285, 2019.
- [38] N. Madi and H. S. Al-Khalifa, “A proposed Arabic grammatical error detection tool based on deep learning,” *Procedia Computer Science*, vol. 142, pp. 352–355, 2018.
- [39] K. Jensen, G. E. Heidorn, and S. D. Richardson, “Natural language processing: the PLNLP approach,” *Springer Science & Business Media*, vol. 196, 2012.
- [40] I. A. Mel’cuk, *Dependency Syntax: Theory and Practice*, SUNY press, Albany, NY, USA, 1988.
- [41] J. Nivre, “Dependency grammar and dependency parsing,” *MSI Report*, vol. 5133, no. 1959, pp. 1–32, 2005.
- [42] R. McDonald, F. Pereira, K. Ribarov, and J. Hajč, “Non-projective dependency parsing using spanning tree algorithms,” in *Proceedings of the HLT/EMNLP 2005 - Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 523–530, Vancouver, BC, Canada, October 2005.
- [43] J. Lafferty, D. Sleator, and D. Temperley, “Grammatical trigrams: a probabilistic model of link grammar,” *Journal of the Chinese Language Computer Society*, vol. 56, 1992.
- [44] D. D. K. Sleator and D. Temperley, “Parsing English with a link grammar,” 1995, <https://arxiv.org/abs/cmp-lg/9508004>.
- [45] م. العثيمين، شرح الأرومية، مكتبة ابن رشد، p. 1435.
- [46] 2020 “Madamira -Arabic Analyzer-.” [Online]. Available: <https://camel.abudhabi.nyu.edu/madamira/>. [Accessed: 06-Jun-].
- [47] Y. Belinkov, A. Magidow, A. Barrón-Cedeño, A. Shmidman, and M. Romanov, “MADAMIRA: a fast, comprehensive tool for morphological analysis and disambiguation of Arabic,” in *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, pp. 1094–1101, Reykjavik, Iceland, May 2014.