A Machine Learning System for Distinguishing Nominal and Verbal Arabic Sentences

Duaa Abdelrazaq[1], Saleh Abu-Soud[2], and Arafat Awajan[1]
[1]Department of Computer Science, Princess Sumaya University for Technology, Jordan
[2]Department of Software Engineering, Princess Sumaya University for Technology, Jordan

Abstract: *The complexity of Arabic language takes origin from the richness in morphology, differences and difficulties of its structures than other languages. Thus, it is important to learn about the specialty and the structure of this language to deal with its complexity. This paper presents a new inductive learning system that distinguishes the nominal and verbal sentences in Modern Standard Arabic (MSA). The use of inductive learning in association with natural language processing is a new and an interdisciplinary collaboration field, specifically in Arabic Language. A series of experiments on 376 well annotated (i.e., Gold Standards) Arabic sentences that range from 2 to 11 words, which present simple to complex MSA sentences, have been conducted. The results obtained showed that the proposed system has distinguished nominal and verbal sentences with an accuracy around 90% for 15% unseen sentences, and around 80% for 75% of unseen sentences.*

Keywords: *Arabic language processing; natural language processing; inductive learning, ILA.*

Received February 14, 2018; accepted April 20, 2018

1. Introduction

Machine learning (ML) with respect to Natural Language Processing (NLP) has gained more attention. This is because of the fact that the availability of the increased number of unstructured and semi structured electronic documents.ML can be used to automatically classify and discover patterns from such electronic documents. The increasing association between ML and NLP has been proved that most NLP problems can be viewed as classification problems [4, 11].

Inductive Learning generates general rules from a set of empirical instances; this process specifies learning by example form. Inductive learning algorithms such as [6, 8, 12, 15] are algorithms that generate specific data into general rules. These works prove that induction of rules from observed data is a useful technique for automatic knowledge acquisition.

The term "إعراب" ,<ErAb, (in Buckwalter transliteration) in Arabic designates the primary constructing tool that is used to analyze each word in a sentence and determines if it is grammatically correct. The first step on the <ErAb is to distinguish the nominal and verbal sentences. The objective of our approach is to design a comprehensive system able to distinguish most of nominal and verbal sentence cases. The majority researchers' investigations were revolving in particular cases for nominal or verbal sentences, and in a partial way using rule models as a sub-task in a complete process of linguistic analysis as [5, 9, 14].

2. Ila: The Inductive Learning Algorithm

In this paper, we apply an inductive learning algorithm called ILA [16], to distinguish between nominal and verbal sentences of Arabic language. ILA which is one of family algorithms that constitute different variations of ILA [1, 2, 17], create a set of classification rules from training examples that is considered a supervised learning approach. This inductive algorithm produces IF-THEN rules in an iterative mode from a set of examples that has a single class. The algorithm depends on the generality of the database patterns and eliminates the unnecessary conditions. When a rule is being found, ILA marks the examples that covered the detecting rule and removes those examples from the training set. ILA rules are more simple and general than those obtained from other known algorithms, as it has been proven in their work, in which the classification capability of ILA increases by the generality of the rules. More discussion and illustrations about ILA can be found in [1, 2, 16, 17].

3. System Framework

To apply machine learning on Arabic Text we have to transfer the unstructured text form to Tags structure database form.

The System framework passes through three main phases: Arabic sentences collection, pre-processing and Classification. This framework is shown in Figure 1.

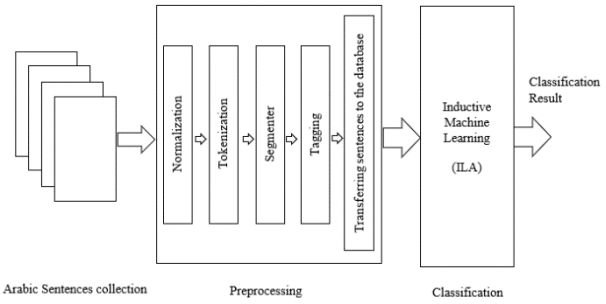

Figure 1. System framework.

3.1. Arabic Sentences Collection Phase

The data set that will be used for building and testing by ILA algorithm consists of 376 sentences, with distinct structures, well-formed and have correct structure.

3.2. Data Pre-Processing Phase

In this phase, sentences are processed and prepared to be used by the classification phase. In order to well understand this phase, it is worthy to mention here that the grammar of Arabic contains the grammar knowledge that is required to analyze a sentence; that is it is important to know the grammar of the sentence in order to understand its meaning. According to Albasria School (المدرسة البصرية), which provided a definition of two sentence types in the traditional Arabic grammar (إعراب) is that nominal sentences (الجملة الاسمية) begin with nominal words (nouns, pronouns, etc.) and verbal sentences (الجملة الفعلية) begin with verbs [7].

In addition, the sentence headed by (ان وأخواتها) also has been considering as a nominal sentence, and the sentence headed by (كان وأخواتها) is considered as a verbal sentence, since this verb is considered to be incomplete (ناقص). For this, we will adopt and follow this common agreement definition for MSA sentences.

Furthermore, sentences can be subdivided into a simple sentence, which is not connected by any means with another sentence, and a compound sentence, which is composed of simple sentences connected with a conjunction article (اداة عطف).

This phase has five main sub-phases; Normalization, Tokenization, Segmentation, Tagging and Transferring to a database as indicated in Figure 1 and illustrated in the following point:

1. *Normalization of the collected sentences*: in this paper, we used un-diacritic normalized sentences dataset. Normalization of words is done by the following choices:
   a. Remove diacritics.
   b. Replace ا أ أ with ا.
   c. Replace يwith ى.
   d. Replace ة with ه.
2. *Tokenization*: in tokenization, the white spaces as well as the punctuation marks are considered as the main markers that separate words in Arabic language. Typically, this process removes the non-Arabic letters, numbers and punctuations. According to our domain which recognizes and handles the sentence as it appears in the text, and since the Penn Tagging is defining all types of words, even the punctuations, we define a sentence as a stream of words that followed by a full stop, question mark, exclamation mark or a comma. Figure 2 presents an example of the tokenization phase.
3. *Stemming-Light- Segmenter*: in addition to affixes, Arabic texts are full of agglutination which contains proclitic (e.g., Articles, prepositions, conjunctions) and enclitics (e.g., Linked pronouns; الضمائر المتصلة) with stems (called lexical forms) [3]. Thus, numerous ambiguities of decomposing textual forms conduct to a significant ambiguity in the part of speech tags of Arabic words.

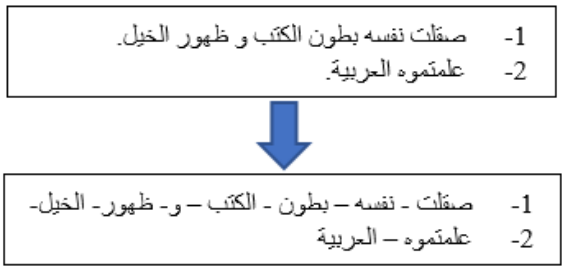

Figure 2. Tokenization example.

As it is mentioned earlier, since our approach appeals the structure of sentences that dedicates to our goal to distinguish the nominal and verbal Arabic sentences, a proper light-segmentation to each word is what matters, to enhance the output accuracy of the Stanford tagging, which uses the Penn Tags [10].

Substantially, each word will be segmented or in-attach any Suffix personal pronouns and Prefix preposition in order to improve the output of Stanford parser tagging set. Thus, the Sentence in Arabic: " صقلت نفسه بطون الكتب وظهور الخيل " ,which is (" Sqlt nfsho bTwn Alktb wzhwr Alzyl " in Buckwalter transliteration), that corresponding in Eng.: (Refined his soul the books and horseback). This sentence will be applied in the preprocessing phase and will be segmented to indicate to any Suffix or Prefix, that have not been recognized in the Stanford Parser Tags set, as illustrated in Figure 3. Hence, the Determiner "ال" is already recognized by the Stanford Parser and it will not be separated.

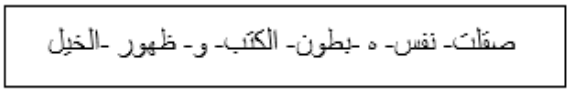

Figure 3. Light- segmenter.

Tables 1, 2, and 3 present linked pronouns, prepositions and the prefix coordinating conjunction in Arabic respectively.

Table 1. The suffix- linked pronouns- in Arabic.

Number	Suffix	Number	Suffix
1	ك	11	ا
2	ه	12	وا
3	ها	13	ن
4	كم	14	ي
5	كما	15	هما
6	نا	16	تا
7	كن	17	ين
8	هم	18	ني
9	هن	19	ان
10	ت		

Table 2. The prefix- preposition- in Arabic.

Number	Prefix
1	ك
2	ل
3	ب

Table 3. The Prefix- coordinating conjunction in Arabic.

Number	Prefix
1	و
2	ف

4. *Part of speech-Tagging*: In The complexity that appears in the Arabic POS tagging is due mainly to different possible decomposition of a word and sentence. Since, a word in Arabic carried an inflection and clitics (e.g., Pronouns, conjunctions, and prepositions) [13]. Thus, to solve the problem that will appear in the accuracy of the Stanford tagging, a segmentation-based approach has been previously adapted in the stemming phase. In this process, after transferring each sentence and their tokens -based segmenting-, Stanford tagging that embedded in the Stanford parser to acquire proper accurate tags will be used. Thus, the Sentence (" SqltnfshobTwnAlktbwzhwrAlzyl " in Buckwalter transliteration), English: (Polished his soul the books and horseback), " صقلت نفسه بطون الكتب وظهور الخيل " will be assigned as the following POS –tags:

Your query

صقلت نفس ه بطون الكتب و ظهور الخيل

Tagging:

صقلت / VBD

نفس/NN

ه/PRP$

بطون/NN

الكتب/DTNN

و/CC

ظهور/NN

الخيل/DTNN

Figure 4. The stanford tags.

Figure 4 presents the Tagged sentence as produced by the online Stanford Parser. We should mention that we use the parser just to get the Penn Tags that have been used and embedded in the parser, and without using the parse tree (the grammatical analysis) of the given sentences.

5. Transferring sentences to examples in the dataset: The representation of text will be modified in order to obtain words-POS-database. Thus, all Tags that correspond to each sentence will be transferred to a row (example) in the dataset. Furthermore, each sentence with its equivalent tags and the decision class of the sentence will be reviewed by Arabic linguistics in order to make sure that the structured sentences will be accuratefor the learning process. Table 4 presents the previous sentence in the database with its equivalent Tags and the decision class.

3.3. Classification Phase

In this phase, in order to classify the MSA sentences to nominal and verbal sentences, we adopt an inductive learning approach by choosing an Inductive Machine Learning -ILA algorithm [16]. The algorithm, as discussed earlier, produces an induction rules from a set of examples by generating IF-THEN rules. Substantially, the main advantage of the algorithm, that the rules are stated in an appropriate form for data exploration as well as the class description.

4. The Proposed System

We will explore ILA in ANLP field for distinguishing nominal and verbal MSA sentences. Typically, the presented supervised learning approach learns automatically from previously annotated sentences using training corpus sentences as a dataset.

Table 4. The equivalent tags.

NO	Tag1	Tag2	Tag3	Tag4	Tag5	Tag6	Tag7	Tag8	Tag9	Tag10	Tag11	Decision
1	VBD	NN	PRP$	NN	DTNN	CC	NN	DTNN	NA	NA	NA	v

4.1. The Training Set

In the training set, the dynamic set of attributes is described in Tags with its own set of possible values, which have been obtained from the original corpus of Arabic POS tags with the parsed Arabic Treebank; these tags have been presented in Table 5.

Table 5. List of Penn POS tags used.

No	Tag	Description
1.	JJ	Adjective
2.	RB	Adverb
3.	CC	Coordinating conjunction
4.	DT	Determiner
5.	FW	Foreign word
6.	NN	Noun, singular or mass
7.	NNS	Noun, plural
8.	NNP	Proper noun, singular
9.	NNPS	Proper noun, plural
10.	RP	Particle
11.	VBP	Verb, non-3rd person singular present
12.	VBN	Verb, past participle
13.	VBD	Verb, past tense
14.	UH	Interjection
15.	PRP	Personal pronoun
16.	PRP$	Possessive pronoun
17.	CD	Cardinal number
18.	IN	Preposition or subordinating conjunction
19.	WP	Wh-pronoun
20.	WRB	Wh-adverb
21.	.	punctuation mark, sentence closer
22.	,	punctuation mark, comma
23.	:	punctuation mark, colon
24.	NOUN_QUANT	Quantifier
25.	ADJ_NUM	Ordinal number

The examples are constructed from a sequence of -Penn POS- tags that represent the sequence of each annotated sentence which will be listed in a table. The rows correspond to the sentences and the columns contain attribute values of the Penn tags correspond to each word in the sentence. Finally, since the sentences' length range are not the same we will indicate to each empty Tags that indicate the end of the sentence with NA attributes, due to the space here reflects a valuable information (i.e., short, long, simple and compound sentences).

4.2. ILA Algorithm

The process of applying a supervised Inductive Machine Learning to a document of Arabic sentences toward distinguishing the nominal and verbal sentences is illustrated in Figure 5.

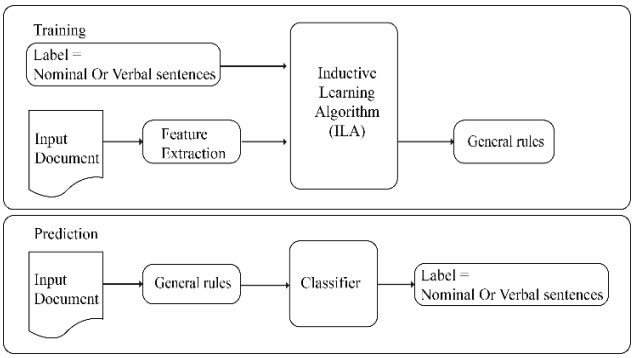

Figure 5. The process of supervised inductive learning.

Once the classified document is labeled by verbal and nominal decisions and entered in the training dataset, features are extracted through applying ILA algorithm in order to extract general rules. Later, the induced rules will be applied on the test set in order to predict the nominal and verbal label decision for the unseen examples. The dataset contains of training examples, considering E, each example, composed of A attributes and a class attribute C with two possible decisions (i.e., nominal and verbal) in our case.

4.3. An Illustrative Example

As an illustrative example by implementing ILA operations in our approach, we consider a training set, consisting of Fourteen examples (i.e., E=14) with eleven attributes (A=11) and one decision (class) attribute with two possible values, {n, v}, (C=2). In this example, Tag i, i ϵ {1,2,3,4.5,6,7,8,9,10,11} are attributes of possible values of Penn Tags sets {e.g., VBD, VBP, DTNN, NN, etc.}, corresponding to their sequence located in the sentences. Table 6 presents the authentic sentences before the pre-processing phase, Table 7 represents the sequence of tags corresponding to the sentences given in Table 6.

Since C is two, {n, v}, (C =2), the first step of the algorithm divided the examples in the two sub-tables which are shown in Table 8, One table for each possible value of the class attribute- nominal and verbal- {n, v}.We will initialize the attributes combination count; j=1 in the first sub-table. The list of attribute combinations comprises: Tag i, iϵ{1,2,3,4.5,6,7,8,9,10,11}.

Table 6. Sentences sample example dataset.

1	وهذه الاسباب لم تهيئه للتعلم،
2	ان قضية تعليم النحو العربي للأجانب يشغل عقول المدرسين.
3	درهم وقاية خير من قنطار علاج.
4	خير لك أن تتألم لأجل الصدق.
5	تتضمن عملية التخطيط صياغة الاهداف التدريسية في صورة قابلة للتقويم.
6	ينام عميقا من لا يملك ما يخاف من فقدانه.
7	أبي سافر الى عمّان.
8	قوة السلسلة تقاس بقوة أضعف حلقاتها.
9	دقيقة الألم ساعة وساعة اللذة دقيقة
10	البستان الجميل لا يخلو من الأفاعي
11	انّه يعيش في بيئة غير عربية وفي مجتمع أجنبي.
12	فليس هناك وجه للمقارنة بين صنفين من الطلبة،
13	يحصد الفلاح القمح.
14	ركبت بالأمس زورقا مع أبي.

Table 7. Object classification training set.

No	Tag1	Tag2	Tag3	Tag4	Tag5	Tag6	Tag7	Tag8	Tag9	Tag10	Tag11	Decision
1	CC	DT	DTNN	RP	VBD	PRP	IN	DTNN	NA	NA	NA	n
2	RP	NN	NN	DTNN	DTJJ	IN	NN	VBP	NN	DTNNS	NA	n
3	NN	JJ	NN	IN	NN	NN	NA	NA	NA	NA	NA	n
4	NN	IN	PRP	RP	VBP	IN	NN	DTNN	NA	NA	NA	n
5	VBP	NN	DTNN	NN	DTNN	DTJJ	IN	NN	NN	IN	DTNN	v
6	VBP	NN	WP	RP	VBP	WP	VBP	IN	NN	PRP	NA	v
7	NN	VBD	IN	NNP	NA	NA	NA	NA	NA	NA	NA	n
8	NN	DTNN	NN	IN	NN	JJR	NNS	PRP	NA	NA	NA	n
9	NN	DTNN	NN	CC	NN	DTNN	NN	NA	NA	NA	NA	n
10	DTNN	DTJJ	RP	VBP	IN	DTNN	NA	NA	NA	NA	NA	n
11	RP	PRP	VBP	IN	NN	NN	JJ	CC	IN	NN	JJ	v
12	CC	VBD	DT	NN	IN	DTNN	RB	NNS	IN	DTNN	NA	v
13	VBP	DTNN	DTNN	NA	NA	NA	NA	NA	NA	NA	NA	v
14	VBD	PRP	IN	DTNN	NN	IN	NN	PRP	NA	NA	NA	v

ILA divides the attributes into distinct combinations with j distinct attributes. For the examining combination {Tag1} with its attribute value "NN", it appears in sub-table A but not in sub-table B with five times appearances. As a result, the maximum number of occurrences is "NN" -the first combination- will be called max-combination. The attribute values "RP" and "CC" will not be considered in this step since they appear in both sub-table A and sub-table B. For combination {Tag2} we have "DT, JJ, IN, DTJJ" with same occurrence of one time, for this case any value can consider as the maximum – combination for that attribute. For {Tag3}, we have "NN" with four occurrences and "PRP", "RP" with one occurrence. Consequently, "NN" value will be as the maximum-combination for {Tag3}. Continuing further with the combination for Tag i, i ϵ {1,2,3,4.5,6,7,8,9,10,11}. After examining all combinations, we found that {Tag1} with the attribute value of "NN" will be selected as the maximum of the maximum – combination (i.e., maximum number of occurrences) between all attributes because it has the maximum appearance for all Tags. Thus, since the value of max-combination is recurrent in 3,4,5,6 and 7 rows, those rows will be marked as classified in sub-table A and the following rule (Rule 1) will be extracted:

- Rule 1: If Tag1 = NN => N

These steps will be applied on the remain unmarked examples in sub-table A (i.e., rows 1, 2 and 8). Thus, "DTNN" is the only attribute value of {Tag1}- since we eliminate "RP" and "CC" for being appeared in sub-table B. "DT" and "DTJJ" the attribute value of{Tag2}, and so on, continuing examining the attribute values from Tag3 to Tag11. As it is shown in Table 8, the number of occurrences is the same for all remaining attributes (i.e., each occurring once). Thus, the first occurrence's attribute by default will be selected (i.e., "DTNN" attribute value of {Tag1}).

Rule 2 is appended to the rule set and the eighth row in sub-table A will be marked as classified:

- Rule 2: If Tag1 = DTNN => N

Table 8. Sub-Tables-According to decision classes partitioned.

Sub-Table A												
Example No.												
Old	**New**	**Tag1**	**Tag2**	**Tag3**	**Tag4**	**Tag5**	**Tag6**	**Tag7**	**Tag8**	**Tag9**	**Tag10**	**Tag11**	**Decision**
1	**1**	CC	DT	DTNN	RP	VBD	PRP	IN	DTNN	NA	NA	NA	n
2	**2**	RP	NN	NN	DTNN	DTJJ	IN	NN	VBP	NN	DTNNS	NA	n
3	**3**	NN	JJ	NN	IN	NN	NN	NA	NA	NA	NA	NA	n
4	**4**	NN	IN	PRP	RP	VBP	IN	NN	DTNN	NA	NA	NA	n
7	**5**	NN	VBD	IN	NNP	NA	NA	NA	NA	NA	NA	NA	n
8	**6**	NN	DTNN	NN	IN	NN	JJR	NNS	PRP	NA	NA	NA	n
9	**7**	NN	DTNN	NN	CC	NN	DTNN	NN	NA	NA	NA	NA	n
10	**8**	DTNN	DTJJ	RP	VBP	IN	DTNN	NA	NA	NA	NA	NA	n
Sub-Table B													
Example No.													
Old	**New**	**Tag1**	**Tag2**	**Tag3**	**Tag4**	**Tag5**	**Tag6**	**Tag7**	**Tag8**	**Tag9**	**Tag10**	**Tag11**	**Decision**
5	**1**	VBP	NN	DTNN	NN	DTNN	DTJJ	IN	NN	NN	IN	DTNN	v
6	**2**	VBP	NN	WP	RP	VBP	WP	VBP	IN	NN	PRP	NA	v
11	**3**	RP	PRP	VBP	IN	NN	NN	JJ	CC	IN	NN	JJ	v
12	**4**	CC	VBD	DT	NN	IN	DTNN	RB	NNS	IN	DTNN	NA	v
13	**5**	VBP	DTNN	DTNN	NA	NA	NA	NA	NA	NA	NA	NA	v
14	**6**	VBD	PRP	IN	DTNN	NN	IN	NN	PRP	NA	NA	NA	v

For the remaining rows, first and the second unmarked remaining rows will be induced rules as follow:

- Rule 3: If Tag2 = DT => N
- Rule 4: If Tag3 = NN => N

By marking the first and the second remaining rows, the whole first sub-table -which indicates the nominal sentences- is now classified, we will progress the same steps on sub-table B.

In the second sub table, "VBP" is the attribute value of {Tag1} appears three times in the first, second and fifth rows in sub-table B. Thus, the three rows will be classified, and Rule 5 will be added to the rule list.

- Rule 5: If Tag1 = VBP => V

Now, we have row 3, 4 and 6, but regarding the algorithm, we will exclude Tag1 for the row 3 and 4, since Tag1 in these rows appeared in the first sub-table. Thus, the only choice for {Tag1} is the attribute with the value of "VBD", which appears in row 6, it will be marked as classified and the sixth rule will be extracted:

- Rule 6: If Tag1 = VBD => V

For row 3, the number of occurrences is the same for all remaining attributes (i.e., each occurring once). The algorithm applies and selects the first one by default (i.e., "PRP" attribute value of {Tag2}). Rule 7 will also be extracted and inserted into the rule set:

- Rule 7: If Tag1 = PRP => V

For the last remain row (i.e., row 4), the occurrences are the same for all remaining attributes (i.e., each occurring once). The algorithm applies and selects the first one by default (i.e., "PR" attribute value of Tag3). Thus, the row will be marked as classified and Rule 8 (i.e., the last rule) will be added to the rules list:

- Rule 8: If Tag3 = PR => V

All the rows of sub-table B are marked as classified; the algorithm terminates and exits with the set of rules extracted to this point.

Furthermore, as it has been noticed that the entire previous example does not fit if the max-combination = ϕ. In that case, according to ILA j will be increased by 1. Table IX presents the case of one remaining row (i.e., the first row in Sub-Table D), which each attribute values appear in both sub tables (i.e., max-combination =ϕ). The algorithm will increase j by 1 and generate combinations of 2 attribute, {Tag1 and Tag2}, {Tag1 and Tag 3}, {Tag 1 and Tag 4} and so on, examining all the possibility. Hence, the "CC and PRP" value of {Tag 2 and Tag 3} combination is disregarded because it appears in sub-table C, and this is applied on combinations of the same conditions. The first and second combinations suit the conditions as they both appear in sub-table D but not in sub-table C for the same attributes. Thus, {Tag1 and Tag2} will be selected. Consequently, the following rule is obtained, and the row is marked as classified:

- IF Tag1= CC and Tag2= RP => V

Table 9. The case of row's attributes that appear in both sub tables.

Sub-Table C											
Tag1	Tag2	Tag3	Tag4	Tag5	Tag6	Tag7	Tag8	Tag9	Tag10	Tag11	Decision
CC	NN	PRP	RB	NOUN_ QUANT	NN	NA	NA	NA	NA	NA	n
CC	NN	PRP	VBP	DTNN	IN	DTNN	IN	DTNN	NA	NA	n
CC	DTNN	DTNN	NN	IN	NNP	NA	NA	NA	NA	NA	n
CC	NN	DTNN	NN	PRP	JJ	NA	NA	NA	NA	NA	n
CC	DT	DTNN	VBP	RP	VBP	NN	NN	NN	DTNN	NA	n
CC	DT	IN	RP	PRP	VBP	NN	NN	DTNN	NN	NA	n
CC	IN	NN	DTNN	CC	RP	NN	DTNNS	IN	DTNN	NN	n
CC	NN	DTNN	IN	NN	DTNN	IN	NN	DTNN	NA	NA	n
CC	NN	PRP	NNS	VBP	IN	DTNN	NA	NA	NA	NA	n
CC	DT	DTNN	RP	VBD	PRP	IN	DTNN	NA	NA	NA	n
DTNN	RP	PRP	NN	NA	NA	NA	NA	NA	NA	NA	n
Sub-Table D											
Tag1	Tag2	Tag3	Tag4	Tag5	Tag6	Tag7	Tag8	Tag9	Tag10	Tag11	Decision
CC	RP	PRP	RP	VBP	NN	DTNN	NA	NA	NA	NA	v
CC	VBP	NNP	IN	NN	VBP	NN	PRP	RB	NN	DTNN	v
CC	VBP	RB	DT	IN	NN	DTNN	IN	NN	PRP	DTJJ	v
CC	VBD	IN	NN	NN	DTNN	DTJJ	IN	NN$	NN$	PRP	v

5. Experiments and Results

In this section, we will evaluate the proposed system to assess its accuracy, through a series of experiments on 376 well annotated (i.e., Gold Standards) Arabic sentences that range from 2 to 11 words, which present simple to complex MSA sentences.

This induction system inserts a set of Arabic training sentences as a file of ordering attribute values set, labeled by a decision attribute for each example. The results are created as a set of individual rules for each of the classification decision attributes.

The evaluation of a learning system is being assessed by the accuracy of the classification rules on unseen examples.

The proposed system will load the database sentences in the memory in order to divide it into two sets, training sentences and test sentences. Thus, after the database has been loaded, it will randomly select the test sentences from the dataset as unseen examples. Consequently, the remaining of the dataset will establish the training set in which the algorithm will run and generate the inductive rules. Subsequently, the unseen examples will obtain and predict the decisions class label. For enhancing the generality of the evaluation results, the test had been randomly conducted on the exceeding cases four times, containing the training sentences and the unseen sentences as well.

To evaluate the proposed system, four experiments have been conducted; each is repeated four times with different portions and different number of sentences as follows: 125, 188, 250, and 376 sentences respectively. Table 10 shows the obtained results by applying the system on different number of sentences, with different percentages of unseen examples. As shown on the table, we got around 89.47% accurate for 15% unseen examples on 125 sentences, while for 75% unseen examples we got around 74.31% accuracy which is not bad for a small number of sentences. However, it is noticed that the accuracy improves as number of sentences increases. Even that the results are domain-dependent and affected by the randomness of unseen sentences, the best results are almost obtained from 376 sentences. Figure 6 summarizes these results in more clearly.

Table 10. Results obtained for different number of sentences.

% of Unseen examples	Average Accuracy			
	125 sentences	188 sentences	250 sentences	376 sentences
15%	89.47	87.58	87.89	92.63
25%	85.62	85.95	90.47	91.48
50%	75.15	83.19	85.44	88.93
75%	74.31	78.58	78.93	78.72

In another experiment, the precision of nominal and verbal sentences is calculated as it defined by the Equation (1) below.

$$Precision = TP \div (TP + FP) \quad (1)$$

Where TP = True Positive
FP = False Positive

Tables 11 and 12 show the average results of 5 experiments, each is repeated 5 times, for 10% of unseen sentences (i.e., 38 sentences) from the total of 376 sentences for Nominal and Verbal sentences respectively.

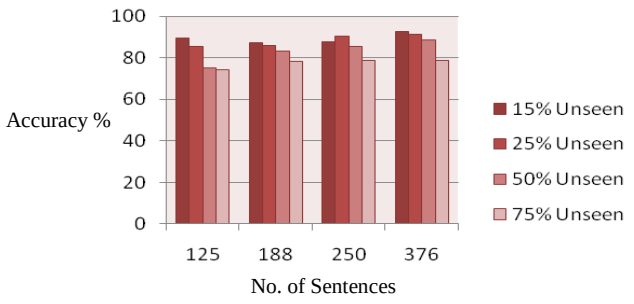

Figure 6. Average accuracy of the system for different number of sentences with different number of unseen examples.

Table 11. Precision for classifying unseen nominal sentences.

Experiment	TP	FP	Precision
1	12	4	%75
2	15	4	%79
3	16	2	%89
4	9	9	%50
5	12	4	%75

Table 12. Precision for classifying unseen verbal sentences.

Experiment	TP	FP	Precision
1	21	1	%95
2	19	0	%100
3	19	1	%95
4	20	0	%100
5	22	0	%100

As it is shown, the system behaves extremely well in predicting the verbal decision than the nominal decision. This is because; from the point of view of Arabic linguists, the nominal sentences basically have a lot of structures than the verbal sentences. Consequently, the nominal decision is more difficult to predict due to their ambiguity structure characteristics.

6. Conclusions

This paper has been carried out with the aim of understanding Modern Arabic sentence structure. The main objective of this work was to implement an inductive machine learning approach to distinguish the nominal and verbal Arabic sentences for MSA. To achieve our goal, ILA as an inductive learning algorithm induced a set of rules based on the training set from a structured data text. Therefore, the sentences have been pre-processed, and structured, through tagging the words sequence in sentences using the Stanford tagger that impeded in their parser. The accuracy of Stanford tags in their online parser tool is not appropriate for the learning process nor have appropriate segmentation. For this reason, in our framework, we produced a semi segmenter to separate any clitic that indicates to any subject or object. Furthermore, the sentences have been rechecked and corrected by Linguists in the stemming and in the tagging phase to obtain Gold standard sentences. The generated rules deal with different sentence structures, word agreement and ordering problem. The results shown in Table 10 proved that proposed system has provided a very good accuracy results with 92.63 in 15% unseen sentences. Furthermore, we have measured the precision of the nominal sentences decisions, as well for the verbal sentences in the unseen test set. The results, as shown in the Tables 11 and 12, illustrate that the system is more accurate in predicting the verbal decision than the nominal decision. According to the point view of Arabic linguists for that case, the reason is that the nominal sentences basically have many structures than the verbal sentences. Finally, the successful results of this research show that ILA is a general and robust algorithm for applying it in the field of ANLP.

References

[1] Abu-Soud S., "A Disjunctive Learning Algorithm for Extracting General Rules," *Journal of Institute of Mathematics and Computer Science (Computer Science Series)*, vol. 10, no. 2, pp. 201-217, 1999.

[2] Abu-Soud S. and Haj Hassan M., "A Parallel Inductive Learning Algorithm," *AMSE journal*, France, 2000.

[3] Awajan A., "Keyword Extraction from Arabic Documents using Term Equivalence classes," *ACM Transactions on Asian and Low-Resource Language Information Processing*,vol. 14, no. 2, 2015.

[4] Daelemans W., Weijters T., and Van den Bosch A., "Empirical Learning of Natural Language Processing Tasks," *Machine Learning: Proceedings of ECML-97*, Springer.

[5] Ditters E., "A Formal Grammar for the Description of Sentence Structure in Modern Standard Arabic," *in Proceedings of Arabic Language Processing: Status and Prospects*, Nijmegen, 2001.

[6] Forsyth R, *Machine Learning principles and techniques*, Chapman and Hall, 1989.

[7] Habash N., Gabbard R., Rambow O., Marcus M., and Kulick S., "Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features," *in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, pp. 1084-1092, 2007.

[8] Hancox P., Mills W., Reid B., *Keyguide to Information Sources in Artificial Intelligence/ Expert Systems*, Cambridge University Press, 1990.

[9] Hammadi O. and Aziz M., "Grammatical Relation Extraction in Arabic Language," *Journal of Computer Science*, vol. 8, no. 6, pp. 891-989, 2012.

[10] Maamouri M. and Bies A., "Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools," *in Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Geneva, pp. 2-9, 2004.

[11] Magerman D., "Statistical Decision Tree Models for Parsing," *in Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, Cambridge, pp. 267-283, 1995.

[12] Michalski R, A theory and Methodology of Inductive Learning, *Artificial Intelligence*, Elsevier, 1983.
[13] Mohamed E. and Kübler S., "Arabic Part of Speech Tagging," *Natural Language Engineering*, vol. 18, no. 4, pp. 521-548, 2011.
[14] Othman E., Shaalan K., and Rafea A., "Towards Resolving Ambiguity in Understanding Arabic Sentence," *in Proceedings of International Conference on Arabic Language Resources and Tools*, 2004.
[15] Quinlan J., *Induction, knowledge and expert systems, in Artificial Intelligence Developments and Applications*, Elsevier Science Publishers, 1988.
[16] Tolun M. and Abu Soud S., "ILA: an inductive learning algorithm for rule extraction," *Expert Systems with Applications*, vol. 14, no. 3, pp.361-370, 1998.
[17] Tolun M., Uludag M., Hayri S., and Abu-Soud S., "ILA-2: An Inductive Learning Algorithm for Knowledge Discovery," *Cybernetics and Systems: an International Journal*, vol. 30, no. 7, pp. 609-628, 1999.

Duaa Abdelrazaq has a master degree in computer Science from Princess Sumaya University for Technology (PSUT). Her research interest is in the area of Artificial Intelligence, Machin learning, Data Mining and Natural language processing. Worked as a teacher at the ministry of education of Jordan between 2005-2016. Working now at United Arab Emirates ministry of education as CDI Teacher.

Saleh Abu-Soud is an associate professor at the Department of Software Engineering in Princess Sumaya University for Technology (PSUT). He got his PhD in Computer Science in 1992 (METU), M. Sc. in Computer Science in 1988 (METU), and B.Sc. in Computer Science in 1985 (Yarmouk University). He was working in Jordan University in the period between 1992 and 1995, then he joined PSUT till now, in which he served as the head of the department of Computer Science in the period from 2005 to 2007. He left to work in NYIT for four years in the period from 2007 to 2011, in which he was a professor of Computer Science and the director of accreditation and quality assurance department in the period from 2007 to 2010. His research interest is in the area of Artificial Intelligence. He is the owner of ILA inductive learning algorithm. He is interested mainly in many research topics as Machine Learning, Biometric Keystroke Dynamics, and Speech Synthesis with inductive learning. He has many research papers and 2 books. He supervised dozens of master students and many PhD students; more details can be seen on (*http://scholar.google.com/citations?user=YjZiOScAAAAJ&hl=en*) and *https://www.researchgate.net/profile/Saleh_Abu-Soud*. He is a member of many international projects.

Arafat Awajan is a full professor at Princess Sumaya University for Technology (PSUT). He received his PhD degree in computer science from the University of Franche-Comte, France in 1987. He held different academic positions at the Royal Scientific Society and Princess Sumaya University for Technology. He is currently the vice president of Princess Sumaya University for Technology and the head of the Human Resources Committee. He was appointed as the chair of the Computer Science Department (2000-2003) and the chair of the Computer Graphics and Animation Department (2005-2006) at PSUT. He had been the dean of the King Hussein School for Information Technology from 2004 to 2007, the Dean of Student Affairs from 2011- 2014, the director of the Information Technology Center in the Royal Scientific Society from 2008-2010 and the dean of the King Hussein School for computing Sciences from 2014 t0 2017.