An Enhanced Twitter Corpus for the Classification of Arabic Speech Acts

Majdi Ahed[1], Bassam H. Hammo[2], Mohammad A. M. Abushariah[3]
Department of Computer Science[1]
Department of Computer Information Systems[2, 3]
King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan[1, 2, 3]

***Abstract*—Twitter has gained wide attention as a major social media platform where many topics are discussed on daily basis through millions of tweets. A tweet can be viewed as a speech act (SA), which is an utterance for presenting information, hiding indirect meaning, or carrying out an action. According to SA theory, SA can represent an assertion, a question, a recommendation, or many other things. In this paper, we tackle the problem of constructing a reference corpus of Arabic tweets for the classification of Arabic speech acts. We refer to this corpus as the Arabic Tweets Speech Act Corpus (ArTSAC). It is an enhancement of a modern standard Arabic (MSA) tweet corpus of speech acts called ArSAS. ArTSAC is more advantageous than ArSAS in terms of its richness of annotated features. The goal of ArTSAC is twofold: Firstly, to understand the purpose and intention of tweets which act in accordance with the SA theory, and hence positively influencing the development of many natural language processing (NLP) applications. Secondly, as a future goal, to be used as a benchmark annotated dataset for testing and evaluating state-of-the-art Arabic SA classification algorithms and applications. ArTSAC has been put in practice to classify Arabic tweets containing speech acts using the Support Vector Machine (SVM) classification algorithm. The results of the experiments show that the enhanced ArTSAC corpus achieved an average precision of 90.6% and an F-score of 89.6%. Substantially it outperformed the results of its predecessor ArTSAC corpus.**

Keywords*—Arabic speech acts; twitter; modern standard Arabic; speech act classification

I. Introduction

People discuss different issues and topics on twitter throughout their tweets. Recently, twitter has gained great attention and attraction from the popular press and, increasingly, from scholars. Speech Act (SA) is an utterance (i.e. a spoken word, statement, or vocal sound) that can be used to present information and also to carry out actions. The idea of a SA can be captured by emphasizing that "by saying something, we do something" [1]. For example, when you ask someone to do something in a sentence like: "Please be quiet"; your utterance represents a request SA. Speech Act Classification (SAC) is the task by which a certain utterance is assigned to a certain predefined SA label such as: assertion, request, etc. based on the content of that utterance. SAC is a traditional classification problem similar to the problem of text classification. Topics that are usually discussed on tweeter represent the subjects of the tweets. These topics are classified into three main types: [2].

1) Entity-oriented topics: Topics about different entities such as famous people (e.g. King Hussein of Jordan), or famous restaurants (e.g. Pizza Hut).

2) Event-oriented topics: Topics about different events and occasions around the world. They are usually about breaking news (e.g. parliament elections in Jordan).

3) Long-standing topics: Topics that are continually discussed on twitter, such as weather, movies, or sports.

Speech Act Theory (SAT) is a linguistic theory that was introduced to formalize speakers' intentions and put them into perspective [3]. SAT aims to understand the utterance defined in terms of a speaker's intention and the effect it has on a listener.

Twitter is one of the big-data sources found on social media. It has hundreds of millions of users who generate around 500 million tweets per day [4]. Due to the tremendous volume of tweets, the problem of classifying and extracting useful information out of them is actually a sort of managing big data. This task can be viewed as a major concern to the field of Data Mining (DM). DM uses different approaches such as classification, association rules, or clustering techniques to discover knowledge in big data.

Defining a catalog of labels (classes) and predicting the label of any given instance based on this catalog is the main goal of classification algorithms. Training a computer machine to classify and label speakers' intentions (retrieved from their utterances) could be viewed as a traditional classification problem. For the problem we are attempting to solve in this study, a catalog of speakers' intentions such as requests, questions, promises, threats, etc. is defined. Then, a classification algorithm is used to discover the speaker's utterance. Such automated utterance classification could be handy in tasks like polarity or sentiment analysis of speakers on social media.

Tweets are usually delivered in a natural language. This fact shows that one of the joint research fields that are heavily indulged in the phenomena of big data is Natural Language Processing (NLP). Generally speaking, a tweet is a short text that usually conveys a single SA. SA classifiers can be used as an initial phase in many contextual mining and NLP tasks such as sentiment analysis, opinion mining, question answering, and rumor detection.

For example, in the case of rumor detection on a social media platform such as Twitter; a SA classifier is needed to

classify different tweets and select the ones that might have rumors. Due to the fact that tweets are microblogs (traditionally 140-character per tweet), they make a good source for SAs classification.

Discovering the SA of tweets could also be used in various NLP tasks such as customer polarity. For instance; assume a company wants to measure the degree of satisfaction of its customers about a certain product such as a new mobile phone. Posts on such a product could be in tens or hundreds of thousands. Manual measurement of customer satisfaction in such situations is very hard and time-consuming; hence the existence of an automated approach to accomplish this task could be very helpful.

Arabic is the fifth widely used language in the world. It is the native language of more than 400 million people. Arabic scripts come in three forms: Classical Arabic; like the holy Quran verses, Modern Standard Arabic (MSA) such as everyday formal press statements or news announcements, and Colloquial Arabic like the native dialect of different Arabic countries [5]. In this paper, we are focusing on MSA language which is a formal language that is understood across all Arabic countries. MSA is a light form of classical Arabic that uses only a well-known and common vocabulary. It maintains a formal but simple and easy-going form. Although classification of speech acts is an active research area for the English language [6, 7, 8]; however, there seems to be a little work done on similar research for Arabic language [9].

The importance of this study is driven by the following facts:

1) SA classification can be used to understand the purpose and the intention behind people's tweets. Knowing the SA behind a tweet could allow us to comprehend the mental and emotional state of the tweeters. Predicting the type of SA of tweets about a certain topic can reveal a lot about people's perspectives or attitudes about that topic. For example, a lot of tweets asking about a certain topic reveal that people are confused about that topic or they are mad and demanding actions about it.

2) Classification features for the English language may not be the same for the Arabic language. It is known that different languages rely on different syntax and semantic characteristics to extract the SAC features. This does not eliminate the fact that some features, such as the question mark at the end of a sentence, represent a cross-lingual extraction feature that classifies such a sentence as a question regardless of the language of the sentence (i.e. universality of SA theory).

3) NLP tasks such as sentiment analysis [10], rumor detection [11], and evaluation of customer satisfaction are important in many online applications today; especially in big data environments where the need for automated tools is urgent.

4) NLP research oriented towards Arabic text is limited [12, 13] and, hence there is a dire need for general purpose Arabic language pre-processing tools and benchmark annotated corpora. The proposed classifier and annotated corpus could be of good value in this regard.

In this paper, we tackle the problem of creating a reference corpus of Arabic tweets for the classification of Arabic speech acts. We refer to this corpus as the Arabic Tweets Speech Act Corpus (ArTSAC). It is an enhancement of a modern standard Arabic (MSA) tweet corpus of speech acts called ArSAS [33]. ArTSAC is more advantageous than ArSAS in terms of its richness of annotated features. The goal of ArTSAC is twofold: Firstly, to understand the purpose and intention of people's tweets which comply with the SA theory, and hence positively influencing the development of many Arabic NLP applications. Secondly, as a future goal, to be used a benchmark annotated dataset for testing and evaluating many Arabic SA classification algorithms and applications.

The remaining of the paper is organized as follows: Section II presents the related work to be used to develop a solution for the aforementioned problem. Section III gives a detailed description of the modified corpus. Section IV discusses the development of the classifier and provides an evaluation of its results. Finally, Section V concludes the work and draws a roadmap for the future work.

II. Related Works

We will limit our literature review to automated SA classifiers developed for English and Arabic languages. Many automated SA classifiers for the English language exist, some are dedicated to Twitter. The earliest attempts to build automated classifiers were oriented towards emails.

An SA classifier for emails and Internet forums was presented in [14]. The authors aimed to use the SAs of an email to identify the intentions of its sender. For example, a simple reply in the e-mail's subject field could indicate a reply to a previous request or a question.

In [15] the authors also worked on emails SAs. They demonstrated that the contextual features of an email can improve that email's SAC. In other words, the syntax and semantic features of an email's text can be used to classify an email. The concept of ontology to classify emails according to the sender's intention was introduced by [16]. The proposed ontology consisted of nouns and verbs that could indicate certain intentions. Applying the ontology produced good results for some nouns and verbs. One drawback of this study was the small size of the proposed ontology and its limitation to simple nouns and verbs.

In [17] the authors developed an annotated SA classifier for the classification of online German discussions. They used an n-grams approach to extract the features. The authors achieved better results with similar previous work. An online chat SA classifier was introduced in [18]. In this work, the authors argued that the first few words in each chat were very predictive of its SA category. They believed that the hearer usually infers the speaker's intention after hearing only a few words of the speaker's utterance. For example, a polite request utterance usually contains the word "please" among its first few words. However, we believe that the works of [17] and [18] neglected the role of discourse and speakers' expectations which are very important in an online chat system. In other words, the expected SA of an utterance is affected by the SA of its previous utterance in a conversation. Hence, it is obvious

that online chats resemble conversations with a discourse. For instance, the expectation after someone greets someone else is to hear a greeting reply. Similarly, after a question, an answer is expected.

An automated SA classifier for educational games was introduced in [19]. The authors argued that the SA taxonomy should be established by using subject matter experts. They believed that a small set of well-defined SA categories were better than many sophisticated categories and that balanced data sets could be misleading. Also, they argued that the data set should be tailored according to real-world applications because the real data set that a classifier may run on in the future may be unbalanced. Their experiments showed no conclusive results for their last assumption regarding data set the balance.

The work of [20] brought attention to Twitter. SA recognition from tweets is considered a classification task. Thus, the primary work was to find a set of robust features appropriate for solid classification. They argued that SAs provide good insights into the communicative behavior of tweeters on Twitter. Again, one of the problems found in this paper is the lack of a benchmark annotated data set as the authors labeled and used their own tweets. The work of [21] was a continuation of their previous work described in [20]. In this work, the authors enhanced the annotated data set and used different classification algorithms than the ones they used earlier for the purpose of comparison.

A new SA classifier was developed by [22]. A new annotated dataset of tweets was processed and constructed. In this study, an enormous number of features (nearly 2000) were extracted and processed. What made this possible was the availability of many pre-processing tools that helped in automatically defining and extracting those features.

In [23], the author proposed a SA analysis of celebrities' tweets. The study showed that celebrities talked to different audiences using different SAs. In his study, the author used the CMC SA taxonomy, which contains 16 categories of SAs [24]. However, such a fine categorization could be problematic for the classifier. The author reported that few SAs did not appear in any tweet.

An automated jihadist messages' detector for twitter was introduced in [25]. In this work, no manual annotation was used. This was because radical tweets used to train the classifier were taken from known jihadist accounts, and those tweets were presumed radical based on their radical tweeters. This form of assumption could tailor or overfit the classifier for certain features. These features could be person stylistic or not broad enough to generalize. We believe so because the result obtained by the classifier were remarkably high (from 89% up to 100%) depending on the dataset. This does not agree with the modest result obtained by other research discussed in our review.

With respect to Arabic SA classification, [26] pointed out that the work in this field is very humble. Here we present a few related studies. In [27], the authors reported on an experimental study of manual annotation of around 400 newspaper sentences. They were processed using two classification algorithms to produce an SA classifier. In their work, they used techniques such as part-of-speech tagging, named entities, and utterance initial words. What was noticeable in this work that the size of the dataset was very small, and the dataset was not representative; some SA classes have many more instances when compared to other classes, so the data set was considered unbalanced. In addition, a single annotator was used in the experiment, usually, more annotators are needed, and an annotation policy should be used.

Another simplified Arabic SA classifier had been described in the studies of [28] and [29]. In their work, the classifier only focused on classifying questions and non-questions utterances. The classifier was used in a conversational agent called ArabChat in order for the agent to determine questions and answer them appropriately. The proposed agent processed the user's utterances through pattern matching and compared them to predefined patterns which represent different topics.

Many research works based on manual non-automated SAs classifications for classical Arabic scripts had been described in [30, 31, 32]. It was argued in these studies that certain SA frequencies may increase depending on the communicative nature of the discourse under study. Hence, SAs classifications cannot be performed in a complete context-free manner without taking into consideration the situation in which the speaker uttered his words.

In [33], the Arabic SA and Sentiment corpus (ArSAS) was described. The corpus contained a set of around (21,000) MSA tweets. Each tweet in the corpus was annotated with an SA label and a confidence factor of annotation for that label. The availability of a specialized corpus such as the ArSAS can highly advance the research in Arabic SAs. The work of [34] is such an example. In this work, the authors developed an Arabic SA classifier for Arabic tweets using both SVM and deep learning algorithms.

From the previous studies we could derive the following conclusions:

1) Researchers are still following the SA taxonomy described in [38]. There was a little variation to tailor the SA taxonomy.

2) There is plenty of room for improvement in SA feature extraction; consequently, an improvement in SA classifications.

3) A benchmark SA corpus to be used across the field is in high demand.

This research modifies the ArSAS corpus of [33] and the work of [34]. The newly constructed Arabic tweets SA corpus (ArTSAC) is richer than ArSAS in terms of introducing new annotations. ArTSAC will be used to train a classification algorithm to classify Arabic SA tweets according to the SA theory and to be used as a benchmark annotated dataset for testing and evaluating many Arabic SA classification algorithms.

III. Proposed Arabic Tweets Act Classifier

Our goal is to create an Arabic SA corpus of tweets rich in annotated features that can be used in classifying SAs,

sentiment polarity, sentiment mining, and other NLP applications. Classification of SAs requires two major components: (1) a reference SA annotated corpus and (2) a suitable classifier. In this section, we discuss in detail the construction of the Arabic Tweets Speech Act reference corpus (ArTSAC) for MSA. Next, we present the Support Vector Machine (SVM) classifier to be used throughout the experiments conducted on the corpus to classify Arabic SA tweets.

A. Construction of the ArTSAC Reference Corpus

The construction of an annotated corpus is an essential step to develop any SA recognition system. The construction of such a corpus is a labor-intensive task. Our proposed ArTSAC corpus for modern standard Arabic SA tweets is a modified version of an open-source SA corpus named ArSAS. The construction of ArTSAC required collecting the Arabic tweets from the ArSAS corpus, extracting all their features and annotating them properly, compiling the list of features, and generating the coded file for the classifier. Fig. 1 illustrates the flow diagram for the constructing the ArTSAC reference corpus. The below subsections are the detailed description of each step towards building the corpus.

1) The collection of arabic tweets: Arabic tweets were obtained from an open-source corpus named ArSAS [33]. It has been developed to experiment with Arabic speech acts and it contains about (21,000) MSA tweets. The tweets of ArSAS were classified according to the SA taxonomy described in [20] and they were organized into one of the following classes/categories:

- Assertions: for example, "سيارتي أسرع من سيارتك" (My car is faster than yours.) It indicates that the speaker commits himself to the truth of what he uttered.
- Expressions: for example, "لقد حزنت لما حدث لسيارتي" (I was sad for what happened to my car.) It indicates an expression of emotion by the speaker.
- Requests: for example, "هل تساعدني في تنظيف سيارتي؟" (Can you help me clean my car?) It indicates a request for service or help made by the speaker.
- Questions: for example, "هل تعلم أين مفتاح سيارتي؟" (Do you know where my car's key is?) It indicates an inquiry about information made by the speaker.
- Recommendations: for example, "يجب أن تستشير الطبيب" (You have to see a doctor.) It indicates advice or recommendation presented by the speaker.
- Miscellaneous: They include different SAs. However, they have relatively few occurrences on Twitter, not enough to warrant a separate category.

A one-to-one association between each tweet and one of the SAs categories was already maintained in the ArSAS corpus. We were careful to make sure that each SA category has enough instances (i.e. tweets) to allow us to robustly define their features. Table I lists the number of tweets for each SA category.

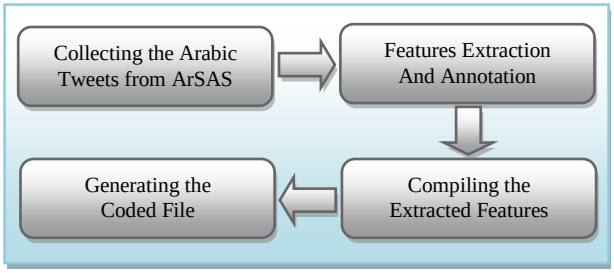

Fig. 1. The Flow Diagram for the Construction of the ArTSAC Corpus.

TABLE. I. THE NUMBER OF TWEETS FOR EACH SA CATEGORY (FROM THE ARSAS CORPUS [33])

Tweet's Class	Number of Tweets
Assertion	8203
Expression	11689
Request	183
Question	749
Recommendation	107
Miscellaneous	60
Total	**20991**

2) Features extraction and annotation: The second step in the development process of the ArTSAC corpus was to extract all proper features from the tweets. Features are the pieces of information or properties within the tweets' messages that convey speech acts. These features represent what a classifier is looking for in order to classify a tweet into one of the aforementioned SA categories listed in Table I.

In order to have the proper guidelines in the feature extraction process of SAs, we conducted a manual analysis of the ArSAS tweets [33] to make sure we have solid insight into the analysis of SAs and their required features.

To conduct the feature extraction task, we got help from two annotators. After we explain to them how to do the features extraction by examples, we performed a pilot task to ensure they understood how to carry out the task. Finally, we could define and extract the following features:

a) Keywords: Some words in a tweet convey certain SA messages. For instance, in an utterance such as "هل بإمكانك مساعدتي رجاءً" (*could you help me please*), the word "رجاءً" (*please*) usually indicates a request SA. This process was an intensive manual process where we have asked the participants to extract up to eight keywords from each tweet in the ArSAS corpus. After we explained to the participants what they have to do, each participant has produced his own list of keywords. After we aggregated the two lists into one featured keywords list, we obtained 1656 unique keywords. The keywords list mainly included constructs such as proper names and nouns.

b) Twitter special characters: Special characters in tweets might designate certain SAs. For example, a special character that is widely used in Twitter is the hashtag '#'. Usually, it indicates an assertion SA. We extracted these special characters automatically using their Unicode values.

c) Topic Label: Each tweet in the ArSAS corpus already has been annotated with a topic label. Labels include Entity, Event, and Long-Standing topics. To the best of our knowledge, this feature was never attempted before in the classification of SAs.

d) Punctuation marks: Few punctuation marks indicate certain SA categories. For example, the question mark '?' usually signals a question or a request SA. Punctuation marks were extracted automatically from the ArSAS tweets corpus.

e) N-grams: Basically, a textual *n*-gram is a sequence of contiguous *n* words that usually co-occur together. N-grams are commonly used in many NLP applications and they usually can help in conveying certain SA messages. For example, the phrase "أﻻ تعتقد" (*Do you think*) usually indicates a question SA. To extract *n*-grams from the tweets in the ArSAS corpus, we perform manual *n*-gram selection with the help of the participants. Each annotator has selected up to 6 possible *n*-grams for each tweet. No limitation was applied to the size of the *n*-gram segments as many *n*-grams represent verses from the holy Quran, popular quotes, or idioms that could span the entire content of some tweets. However, most of the extracted *n*-gram features were *bi*-gram and *tri*-gram. Other possible segments were 4-grams and 5-grams. Finally, the compiled lists of the annotators have been aggregated into one list of 2658 unique *n*-gram features.

f) Emoticons: Expressing emotions through icons are widely used in social media. Emoticons expressing happiness, sadness, etc. are highly informative in reflecting tweeters' attitudes and moods; hence they can convey certain types of SAs. We automatically extracted emoticons from the ArSAS tweets and compiled a list of 68 emoticons.

g) Links: Hyperlinks are impeded in many tweets. They point to different locations and they possibly could indicate certain types of SAs. Hyperlinks have defined structure, which made extracting them automatically an easy task.

h) Sentiment label: Every tweet in the ArSAS corpus had been already annotated with a sentiment label (positive, negative, mixed, or neutral). We used the sentiment features in the classification process as they may convey certain SAs such as recommendations or assertions. Up to our knowledge and from the literature, the sentiment features have never been attempted in the classification problem of SAs.

i) Tweet's length: Tweets are varying in length. Usually, there is a correlation between the tweet's length and the SA within the body of the tweet. Our analysis of ArSAS showed, for instance, that an expression tweet is usually longer in size than a request tweet. Accordingly, for this feature, we assumed that a long tweet is one that has more than 50 characters; otherwise, it is considered a short tweet.

At the end of the feature extraction task, we could draw the following conclusions:

- Feature extraction was performed automatically and manually. The automatic task was the easiest. It has been applied to extract well-defined features such as special characters and emoticons. For automatic annotation, a set of tools was developed. Each tool was used to extract specific features as discussed earlier. The following pseudo-code is a generalized form of the algorithm LookupTableConstructor. This table is accessed by all tools to construct the ArTSAC corpus. Table II shows a sample of the generated features in the LookupTable.

Algorithm: LookUpTableConstructor( )
Pre-request: features
***Process*:**
while there are more tweets
read a tweet's feature from ArSAS
if the feature is not null, then
search for the feature in the feature's LookupTable
if not exist, then
add a feature to the last location in feature's LookupTable
end if
end if
end while
***Results*: feature's LookupTable**

- The manual feature extraction task was conducted by annotators through processing 21,000 tweets from the ArSAS corpus. Although the manual analysis was an intensive task, it was essential to get an in-depth understanding of the characteristics and different usages of SAs. The manual process was used to extract five features.

- Only annotations that have been agreed upon by both annotators have been aggregated and included in the features lists of our ArTSAC corpus. Table III shows a summarization of the extracted features from the Arabic tweets and their corresponding counts.

3) Compiling the extracted features: The extraction of the features was followed by the coding step. To assist the automatic coding of a feature, we developed a LookupTable for each feature. The LookupTable is a binary table contains a unique occurrence of all possible values of that feature extracted from the Arabic ArSAS tweets. Each LookupTable is built by scanning its corresponding column(s) in the corpus and adding a unique occurrence value for all possible values of that feature.

To facilitate this final process, we developed a Graphical User Interface (GUI) to manage the compilation of each feature extracted from the tweets and assigning SAs to tweets. Table IV lists the different functions performed by the system and the numbers of the extracted features. The values of features are binary values located from the LookupTable. A value of 0 means that the feature does not exist in a tweet, otherwise it is 1. Fig. 2 depicts the GUI functions to be used to compile the extracted features into the final DataFile.csv.

TABLE. II. A SAMPLE OF THE FEATURES IN THE LOOKUPTAPLE EXTRACTED FROM THE ARABIC TWEETS

Keywords Features	Initial Words Features	Punctuations	Special Characters	N-Gram Features	Emoticons Features	Speech Acts	Topic	Sentiment Label
غانا	المباراة القادمة	"	#	كأس العالم	😣	Assertion	Event	Positive
شرم الشيخ	هل هذه	?	❗	شباب العالم	😢	Expression	Entity	Negative
تيران	وزير خارجية	!	✨	بسم الله	😊	Request	Long-Standing	Neutral
مصر	ومع السيسي		👎	افصل لاعب	😵	Question		Mixed
اوروبا	اهداف المباراة		⚽	حصار قطر		Recommendation		
الجزائر	طبعا 25		❤	ولي العهد		Miscellaneous		
محمد صلاح	منتدى		🏃	تفتح تحقيق				
قطر	يعني رايح		❓	المزيد من				
سويسرا	قولوا_ل_قطر_كبتين		✊	ثورة يناير				
مرتضى منصور	بسم الله		🏆	حصار اقتصادي				
الزمالك	هذا الربيع		👍	وزير خارجية				
الدنمارك	ملابس		🤦	النجم المتالق	😊			
مصر	دول المال		👂	الدوري الانجليزي				
هدف	رحم الله		💔	الربيع العربي	😔			
السيسي	تصفيات كأس		💋	ببيعوا سمك	😳			
شكرا	ياه جه		👐	العالم العربي	🌍			

TABLE. III. SUMMARIZATION OF THE EXTRACTED FEATURES AND THEIR COUNTS EXTRACTED FROM THE ARABIC TWEETS

Feature Name	Count of Features
Punctuation	3
Twitter Special Chars	172
Topic	3
Sentiment	4
Emoticons	68
Keywords	1656
N-grams	2658
Speech Act categories	6
Link	1
Long	1

TABLE. IV. THE SYSTEM'S MAIN FUNCTIONS AND EXTRACTED FEATURES

Function	What it does	Number of features
Keywords Coding	Assigning values to keyword features	1656
Characters Coding	Assigning values to character features	172
Topic Coding	Assigning values to the topic features	3
Punctuation Coding	Assigning values to the punctuation features	3
N-Gram Coding	Assigning values to n-gram sequences	2658
Emoticons Coding	Assigning values to the existence of emoticons	68
Link Coding	Assigning values to the existence of hyperlinks	1
Sentiments Coding	Assigning values to the types of sentiments	4
Length Coding	Assigning a value to the length of a tweet	1
Speech Act Coding	Assigning a value to the SA type	6
Save Coded Data	Saving the compiled table of features as (DataFile.csv)	-
WEKA	Launching the Weka's package	-

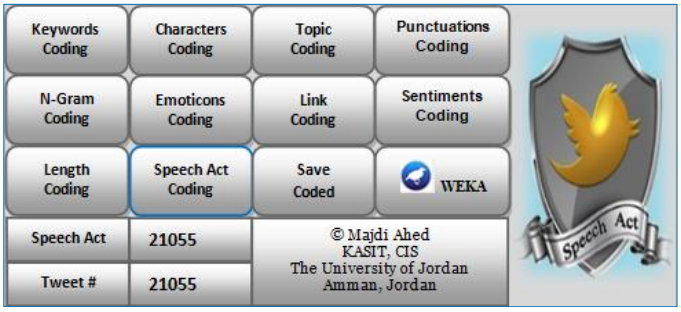

Fig. 2. The GUI of the ArTSAC Corpus.

4) Generating the coded file: The final step in the process of developing the ArTSAC corpus was to generate the SA data file, which is a binary coded file containing all values of SA features. The final file is an Excel comma-separated file (".csv"), suitable to be processed by Weka's SVM algorithm. We called this file "DataFile.csv".

It is important to mention that the structure of DataFile.csv conforms to the structure of the dataset, which would be processed by Weka. This structure has a header of metadata, which is required by Weka to identify each attribute in the file. The header has labels such as *a*1, *a*2, *a*3,..., etc. where '*a*' stands for an attribute and the last column is labeled with '*c*', which contains the value of the SA class (c.f. Table I).

IV. DEVELOPMENT OF THE CLASSIFIER

A. Support Vector Machine

Support Vector Machine (SVM) lies under the category of supervised learning algorithms used for classification. SVM was originally designed to work when data has exactly two classes. In other words, it can be used with binary classification problems. The multiclass SVM problem aims to assign labels to instances, where the labels are drawn from a finite set of several elements.

The traditional approach to solving this problem using SVM is to reduce the single multiclass problem into several multiple binary classification problems. The most common technique in practice is to build one-versus-all classifiers and to choose the class which classifies the test instances with the greatest margin. Another strategy is to build a set of one-versus-one classifiers and to choose the class that is selected by the most classifiers. While this involves building classifiers, the time for training classifiers may actually decrease, since the training data set for each classifier is much smaller.

One way to solve the SVM training problem is to use sequential minimal optimization (SMO) [36, 37]. The setup parameters of SVM were gamma and kernel. Also, we used the C parameter to control the cost of misclassification on the training data. The best performance of SVM was when setting the kernel to "poly", gamma to "auto", and C to 1.

The annotated tweets and the extracted features that we obtained from the previous step were used to train the SVM classifier. The data was saved as a single Excel sheet named (DataFile.csv). In this research, we used Weka (Waikato Environment for Knowledge Analysis) machine learning software [35], which is developed at the University of Waikato, New Zealand. Weka's SVM was implemented as a Java class that has properties. In this implementation, all missing values were replaced, and nominal attributes were transformed into binary ones. Furthermore, and by default, all attributes were normalized. This means that all output coefficients would be based on the normalized data rather than the original data. Such a step is very important for interpreting the results of the classifier.

For multiclass classification, we used Weka's SVM which implemented a pairwise one-versus-one classification technique. The option that fits calibration models to the outputs of SVM is used to achieve accurate probability estimates. However, the predicted probabilities in the multi-class classification are coupled by using Hastie and Tibshirani's pairwise coupling method [39].

One advantage of using Weka is its flexibility of providing a set of alternatives to perform testing of the created classification model. These alternatives include use training set, supplied test set, cross-validation, and percentage split. In our experiments, we used the training set option to perform the testing, such that the training dataset (DataFile.csv) was also the test dataset. The output of training the classifier is a set of important measures which are: precision, recall, and F-score. Table V shows the results of running the SVM classifier on the ArTSAC dataset.

TABLE. V. THE PERFORMANCE EVALUATION OF THE SVM CLASSIFIER RUNNING ON THE ARTSAC DATASET

SA Category	Precision	Recall	F-Score
Assertion	0.963	0.855	0.894
Expression	0.882	0.966	0.922
Request	1.000	0.112	0.201
Question	1.000	0.065	0.123
Recommendation	0.911	0.809	0.857
Miscellaneous	1.000	0.083	0.154
Weighted Average	**0.906**	**0.903**	**0.896**

B. Evaluation of ArTSAC

Before we discuss the results we obtained from our modified ArTSAC corpus, we start with highlighting the previous results obtained from the ArSAS corpus [33] then we compare the results from running SVM on our modified ArTSAC corpus and compare it with the ArSAS corpus.

1) Features extractions in ArSAS and the modified ArTSAC: The features of ArSAS were extracted using the Farasa part-of-speech tagger [40], which has been modified to extract hashtags, emojis, and URLs. On the other hand, features such as unigrams, bigrams, and trigrams were extracted manually [34]. Our modified ArTSAC made benefits from all features in ArSAS in addition to the enhanced set of extracted features. Wherever applicable, the new features of ArTSAC were extracted automatically. Others were extracted manually. All ArTSAC features were extracted through a developed system as shown in Fig. 2.

2) The results of running SVM on the modified ArTSAC: Table V shows the results of running the SVM classifier on the modified ArTSAC corpus. Here we report the F-Score rate for each SA category. The Expression SA category achieved the highest F-Score with a rate of (92.2%). This was followed by the Assertion SA (89.4%), Recommendation SA (85.7%), Request SA (20.1%), Miscellaneous (15.4%), and Question SA (12.3%). However, the least number of tweets were in the Recommendation category (107 tweets) and the Miscellaneous category (60 tweets). The weighted average of all features achieved an F-Score rate of (89.6%).

3) Comparison between ArSAS and the modified ArTSAC: Here we report the comparison results of the SVM classifier running on the ArSAS dataset and the ArTSAC dataset. In the first experiment, we ran Weka's SVM on ArTSAC using the same feature set of ArSAS [33], which include the following features:

- Lexical features: unigram, bigram, and trigram segments.
- Syntactic features: punctuation marks, twitter special characters, Emoticons, and hyperlinks.
- Structural features: tweet's length, and part-of-speech (POS) tags.

4) In the second experiment, we ran Weka's SVM using all features in ArTSAC. Table VI shows the comparison results of running SVM on both datasets: ArSAS and ArTSAC using the F-Score measure.

Table VI shows that the F-Score rate of running the SVM algorithm on ArTSAC using all compiled features is (89.6%), which outperformed the same algorithm running on the original ArSAS dataset with an F-Score rate of (86.2%). However, when we attempted to run SVM on our ArTSAC dataset using the same features as in the ArSAS dataset, we got an F-Score rate of (81.2%). The reason for getting a lower F-Score rate compared with the original ArSAS dataset (using the same features), could due to the following main reasons: (1) in our study we used Weka's SVM algorithm, while in [34] we

don't know exactly how they implemented their SVM algorithm, and (2) we used SMO to optimize the SVM training set along with tuning parameters (*kernel*, *gamma* and *C*), while in [34] it was not clear what parameters they used to tune their algorithm. Fig. 3 shows the F-Score results of running SVM on different Arabic speech acts datasets extracted from Arabic tweets.

The results in Fig. 3 were achieved by using all features in ArTSAC and picking the tweets that have 0.8 and above confidence scores. Then we directed the SVM classifier to use only four SA classes out of the six classes explained in Table I, which are: Assertion, Expression, Request, and Question. The reason for reducing the number of classes to four was because the number of tweets found under the other two classes was very few (c.f. Table I); Recommendation (107 tweets) and Miscellaneous (60 tweets). Therefore, we believe that they might negatively affect the performance of the SVM classification algorithm. We also believe that the better performance of the ArTSAC dataset was due to two main reasons: (1) The newly added features, mainly the sentiment label, indicated a great deal of association between a tweet sentiment label and its SA category, and (2) The careful manual annotation of the keywords as well as the extended *n*-gram segments (i.e. 4-gram, 5-gram and beyond) added more semantic concentration to the extracted features and took the tweets to a level beyond the bag of words.

TABLE. VI. RESULTS OF RUNNING SVM CLASSIFIER ON BOTH ARSAS AND ARTSAC DATASETS

Test	F-Score
SVM/original ArSAS [34]	0.862
SVM/ArTSAC using same features as in ArSAS [34]	0.812
SVM/ArTSAC using all compiled features in ArTSAC	0.896

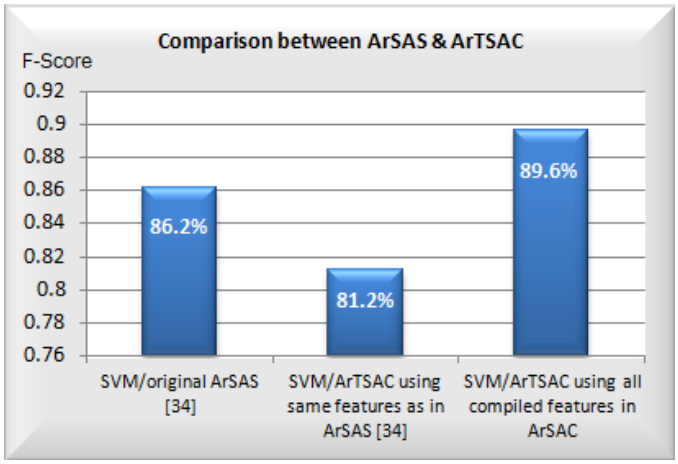

Fig. 3. F-Score Results of Running SVM on ArSAS and ArTSAC Datasets.

V. CONCLUSIONS

In this paper, we presented the development and construction of a richly annotated reference corpus of Arabic tweets for speech act classifications. The corpus, named ArTSAC, was built on top of a previous open-source modern standard Arabic twitter of SA corpus named ArSAS. ArTSAC inherited the features of ArSAS and added more annotated features before it has been put in practice with an SVM classification algorithm to classify Arabic tweets containing SAs.

The goal of ArTSAC is twofold: Firstly, to understand the purpose and intention of people's tweets which act in accordance with the SA theory, and hence positively influencing the development of many online applications. Secondly, as a future goal, to be used as a benchmark annotated corpus for testing and evaluating many Arabic software applications. ArTSAC has been put in practice to classify unseen Arabic tweets containing speech acts using the Support Vector Machine (SVM) classification algorithm. The results from our initial experimentation show that our developed corpus using the SVM algorithm achieved an average precision of (90.6%) and an F-score of (89.6%).

As for future work, we plan to use the ArTSAC corpus with deep learning based model for classifying speech-acts using a convolutional neural network (CNN).

ACKNOWLEDGMENT

The authors would like to thank the editor and the esteemed reviewers for their valuable comments to enhance the readability of the manuscript. Also they would like to thank the anonymous volunteers for doing the annotations and validation of the corpus.

REFERENCES

[1] A. S. Panah and M. M. Homayounpour, "Speech acts classification of Farsi texts," in 2008 International Symposium on Telecommunications, pp. 539-542. IEEE, 2008.

[2] X. Zhao, and J. Jiang, "An empirical comparison of topics in twitter and traditional media," Singapore Management University School of Information Systems Technical paper series. 2011.

[3] J. A. Austin, How to Do Things With Words, 2nd ed., Cambridge, Massachusetts, United States: Harvard University Press, 1975.

[4] X. Liao, Y. Huang, J. Wei, Z. Yu and G. Chen, "A Heterogeneous Graph Model for Social Opinion Detection," in International Conference on Machine Learning and Cybernetics ICMLC, Lanzhou, China, 2014.

[5] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," ACM Transactions on Asian Language Information Processing (TALIP), vol. 8, no. 4, pp. 1-22, 2009.

[6] G. Xu, H. Lee, M. W. Koo, and J. Seo, "Convolutional Neural Network using a threshold predictor for multi-label speech act classification," in 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, South Korea, pp. 126-130. IEEE, 2017.

[7] D. Kim, H. Kim, and J. Seo, "Speech Act Classification Based on Individual Statistical Models in a Multi-Domain," in The 16th IEEE International Symposium on Robot and Human Interactive Communication, Jeju, South Korea, pp. 845-847. IEEE, 2007.

[8] H. Xuefeng. and Z. He, "Methods and characters of speech acts in online shopping," in 2012 IEEE Symposium on Robotics and Applications (ISRA), Kuala Lumpur, pp. 416-418. IEEE, 2012.

[9] F. Al-Hindawi and H. Al-Masudi, "The Speech Act Theory in English and Arabic," Open Journal of Modern Linguistics, vol. 4, pp. 27-37, 2014.

[10] Y. Ren and J. Tian, "Sentiment Analysis of Internet Performance Data," in 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, pp. 622-628. IEEE, 2017.

[11] S. Zamani, M. Asadpour and D. Moazzami, "Rumor Detection for Persian Tweets," in 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, pp. 1532-1536. IEEE, 2017.

[12] W. Alabbas, H. M. Al-Khateeb, and A. Mansour, "Arabic Text Classification Methods: Systematic Literature Review of Primary

Studies," in 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, pp. 361-367. IEEE, 2016.

[13] N. Abdelhade, T. Hassan, A. Soliman, and H. Ibrahim, "Detecting Twitter Users' Opinions of Arabic Comments During Various Time Episodes via Deep Neural Network," in International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, pp. 232-246. Springer, Cham, 2017.

[14] M. Jeong, C.Y. Lin, and G. G. Lee, "Semi-Supervised Speech Act Recognition in Emails and Forums," in 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, vol. 3, pp. 1250-1259, 2009.

[15] V. Carvalho and W. Cohen, "Improving "email speech acts" Analysis via n-gram selection," in HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech, NY, USA, pp. 35-41, 2006.

[16] W. Cohen, V. Carvalho and T. Mitchell, "Learning to Classify Email into Speech Acts," in Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp. 309-316, 2004.

[17] B. Bayat, C. Krauss, A. Merceron and S. Arbanowski, "Supervised Speech Act Classification of Messages in German Online Discussions," in Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, Florida, USA, 2016.

[18] C. Moldovan and V. Rus, and A. C. Graesser, "Automated Speech Act Classification for Online Chat," in The 22nd Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, USA, pp. 23-29, 2011.

[19] V. Rus, C. Moldovan, N. Niraula and A. C. Graesser, "Automated Discovery of Speech Act Categories in Educational Games," in The 5th International Conference on Educational Data Mining Society, Chania, 2012.

[20] R. Zhang, D. Gao and W. Li, "What Are Tweeters Doing: Recognizing Speech Acts in Twitter," in Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, USA, 2011.

[21] R. Zhang, D. Gao and W. Li, "Towards scalable speech act recognition in Twitter: tackling insufficient training data," in Proceedings of the Workshop on Semantic Analysis in Social Media, Avignon, pp. 18-27, 2012.

[22] S. Vosughi and D. Roy, "Tweet Acts: A Speech Act Classifier for Twitter," in The Tenth International AAAI Conference on Web and Social Media, Cologne, Germany, 2016.

[23] D. Nemer, "Celebrities Acting up: A Speech Act Analysis in Tweets of Famous People," Journal of Social Networking, vol. 5, no. 1, pp. 1-10, 2016.

[24] S. C. Herring, A. Das, and S. Penumarthy, "CMC Act Taxonomy," 2005. [Online]. Available: http://info.ils.indiana.edu/~herring/cmc.acts, [Accessed Feb. 17, 2020].

[25] M. Ashcroft, A. Fisher, L. Kaati, E. Omer and N. Prucha, "Detecting jihadist messages on twitter," in European Intelligence and Security Informatics Conference, Manchester, UK, pp. 161-164. IEEE, 2015.

[26] A. A. Elmadany, S. M. Abdou and M. Gheith, "Recent Approaches to Arabic Dialogue Acts Classifications," in The 4th conference of Natural Language Processing, Sydney, Australia, vol. 5, no. 4, pp. 117-129, 2015.

[27] L. Shala, V. Rus, and A. C. Graesser, "Automated Speech Act Classification in Arabic," Subjectivity and Cognitive Processes, vol. 14, no. 2, pp. 284-292, 2010.

[28] M. Hijjawi, Z. Bandar and K. Crockett, "User's Utterance Classification using Machine Learning for Arabic Conversational Agents," in 5th International Conference on Computer Science and Information Technology, Amman, Jordan, pp. 223-232. IEEE, 2013.

[29] M. Hijjawi, Z. Bandar, K. Crockett, and D. Mclean, "ArabChat: An Arabic Conversational Agent," in 6th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, pp. 227-237. IEEE, 2014.

[30] ElA'awar, "Verbal Actions in Surat Al-Kahf - A Deliberative Study," Department of Linguistics, Faculty of Arts and Sciences, University of Mentor, Algeria, Algeria, 2011.

[31] M. Medawar, "Verbal Actions in the Holy Quran (Surat Al-Baqara)- A Deliberative Study," Department of Arabic Language, College of Arts and Sciences, Hajj Lakhdar University, Algeria, Algeria, 2014.

[32] F. A. M. Jawad, "A Pragmatic Analysis of Illocutionary Speech Acts in Standard Arabic with a Special Reference to Al-Ashter s 'Epistle'," Journal of University of Babylon 19, no. 4, pp. 606-625, 2011.

[33] A. Elmadany, H. Mubarak and W. Magdy, "ArSAS: An Arabic Speech-Act and Sentiment Corpus of Tweets," in 3rd Workshop on Open-Source Arabic Corpora and Processing Tools, Miyazaki, Japan, p. 20, 2018.

[34] B. Algotiml, A. Elmadany, and W. Magdy, "Arabic Tweet-Act: Speech Act Recognition for Arabic Asynchronous Conversations," in In Proceedings of the Fourth Arabic Natural Language Processing Workshop, pp. 183-191. 2019.

[35] G. Holmes, A. Donkin and I. H. Witten, "Weka: A machine learning workbench," in Proceedings of ANZIIS'94-Australian New Zealand Intelligent Information Systems Conference, pp. 357-361. IEEE, 1994.

[36] Nabble, "Explanation of SMO Parameters?," [Online]. Available: http://weka.8497.n7.nabble.com/Explanation-of-SMO-Parameters-td21768.html, [Accessed Feb. 17, 2020].

[37] J. Platt, "Fast training of support vector machines using sequential minimal optimization," Advances in Kernel Methods-Support Vector Learning, AJ, MIT Press, Cambridge, MA, pp. 185-208, 1999.

[38] J. R. Searle, Expression and Meaning: Studies in the theory of speech acts, Cambridge University Press, 1985.

[39] T. Hastie, and R. Tibshirani, "Classification by pairwise coupling," in Advances in neural information processing systems, pp. 507-513. 1998.

[40] K. Darwish, and H. Mubarak, "Farasa: A new fast and accurate Arabic word segmenter," in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 1070-1074. 2016.