

# Syntactic- and morphology-based text augmentation framework for Arabic sentiment analysis

**Rehab Duwairi** <sup>Corresp., 1</sup>, **Ftoon Abushaqra** <sup>2</sup>

<sup>1</sup> Department of Computer Information Systems, Jordan University of Science and Technology, Irbid 22110, Jordan

<sup>2</sup> Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan

Corresponding Author: Rehab Duwairi

Email address: rehab@just.edu.jo

Arabic language is a challenging language for automatic processing. This is due to several intrinsic reasons such as Arabic multi-dialects, ambiguous syntax, syntactical flexibility, and diacritics. Machine learning and deep learning frameworks require big datasets for training to ensure accurate predictions. This leads to another challenge faced by researches using Arabic text; as Arabic textual datasets of high quality are still scarce. In this paper, an intelligent framework for expanding or augmenting Arabic sentences is presented. The sentences were initially labelled by human annotators for sentiment analysis. The novel approach presented in this work relies on the rich morphology of Arabic, synonymy lists, syntactical or grammatical rules, and negation rules to generate new sentences from the seed sentences with their proper labels. Most augmentation techniques target image or video data. This study is the first work to target text augmentation for Arabic language. Using this framework, we were able to increase the size of the initial seed datasets by 10 folds. Experiments that assess the impact of this augmentation on sentiment analysis showed a 42% average increase in accuracy, due to the reliability and the high quality of the rules used to build this framework.

# Syntactic- and morphology-based text augmentation framework for Arabic sentiment analysis

Rehab M. Duwairi<sup>1</sup>, Ftoon M. Abushaqra<sup>2</sup>

<sup>1</sup> Department of Computer Information Systems, Jordan University of Science and Technology, Irbid 22110, Jordan.

<sup>2</sup> Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan.

Corresponding Author:

Rehab Duwairi<sup>1</sup>

P.O. Box 3030, Irbid, Irbid, 22110, Jordan.

Email address: [rehab@just.edu.jo](mailto:rehab@just.edu.jo)

# Syntactic- and morphology-based text augmentation framework for Arabic sentiment analysis

Rehab Duwairi<sup>1</sup>, Ftoon Abushaqra<sup>2</sup>

<sup>1</sup> Department of Computer Information Systems, Jordan University of Science and Technology, Irbid 22110, Jordan.

<sup>2</sup> Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan.

Corresponding Author:

Rehab Duwairi<sup>1</sup>

P.O. Box 3030, Irbid, Irbid, 22110, Jordan

Email address: Rehab@just.edu.jo

## Abstract

Arabic language is a challenging language for automatic processing. This is due to several intrinsic reasons such as Arabic multi-dialects, ambiguous syntax, syntactical flexibility, and diacritics. Machine learning and deep learning frameworks require big datasets for training to ensure accurate predictions. This leads to another challenge faced by researches using Arabic text; as Arabic textual datasets of high quality are still scarce. In this paper, an intelligent framework for expanding or augmenting Arabic sentences is presented. The sentences were initially labelled by human annotators for sentiment analysis. The novel approach presented in this work relies on the rich morphology of Arabic, synonymy lists, syntactical or grammatical rules, and negation rules to generate new sentences from the seed sentences with their proper labels. Most augmentation techniques target image or video data. This study is the first work to target text augmentation for Arabic language. Using this framework, we were able to increase the size of the initial seed datasets by 10 folds. Experiments that assess the impact of this augmentation on sentiment analysis showed a 42% average increase in accuracy, due to the reliability and the high quality of the rules used to build this framework.

**Keywords:** Text Augmentation, Sentiment Analysis, Arabic Text, Natural Language Processing, Morphology-Based Augmentation.

## Introduction

Arabic language is considered the most widely spoken language among the Semitic languages [1, 2]. It is also one of the popular languages in the world. As the statistical studies in 2019 mentioned [3], Arabic language is spoken by nearly 319 million people and is ranked the fifth

between the world's languages after Chinese, Spanish, English, and Urdu\Indian. Arabic native speakers are distributed throughout the Arab World as well as many other nearby areas. Also, Arabic has around 30 modern varieties or dialects; one of them is the standard form MSA (Modern Standard Arabic) [4]. In 2012, the United Nations Economic and Social Commission for West Asia (ESCWA) mentioned that Arabic language has achieved the highest growth rate on the Internet compared to other languages. Therefore, recently digital Arabic content on the internet became fairly large. However, this does not deny the reality that Arabic is considered a highly ambiguous language, especially when trying to analyze, classify and process Arabic data automatically.

Recently, many efforts have investigated the Arabic language whether to analyze the text [5, 6, 7], parse statements [8, 9], analyze sentiment [10, 11, 12, 13], recognize speech [14, 15], translate statements [16, 17], or detect depression [18]; all these applications require the existence of comprehensive Arabic datasets. Building a dataset is not an easy task, as it requires tremendous effort, time and cost. Also, the recent application of machine learning and deep learning requires huge datasets which contain billions of records. For example, training a sentiment classifier using deep learning methods requires huge data properly labelled with polarity information. Therefore, an automatic expansion for Arabic datasets is very favorable, especially when knowing that manually collecting and annotating data are troublesome [19].

Sentiment analysis is the task of processing data, mainly textual, in order to determine its polarity, i.e., positive, negative, or neutral [20, 21]. This task has several real-world applications with great impact on important domains such as business [22], politics [23, 24], tourism [25] and marketing [26]. In general, sentiment analysis could be treated as an unsupervised learning task [27], supervised learning task [28], or a hybrid of both. Unsupervised learning for sentiment analysis relies on sentiment lexicons. By comparison, supervised learning requires the existence of annotated or labelled data to train the classifiers. Ideally, data labeling for sentiment analysis mandates that each instance must be assigned a label from: positive, negative, or neutral. This task of labelling is usually human-based and thus it is costly. Corpora for sentiment analysis are usually gathered from social media; and due to the multilingual nature of such media, several researchers directed their efforts towards multilingual sentiment analysis [29, 30, 31].

Deep learning has received unprecedented attention in recent years and provided state-of-the-art results in many fields including sentiment analysis [32, 33, 34, 35, 36]. However, deep neural networks need large amounts of data to train and tune their parameters. Data augmentation is a technique for expanding the datasets, and it was paired with deep learning applications. It has been used successfully with vision data and recently has received attention with textual data. Augmenting data that was initially labelled for sentiment analysis involves generating new sentences relying on the existing ones. The simplest form is to use the synonyms of the words to create new sentences with the same labels as the original ones. In this work, a framework for extending the size of datasets that were originally labeled for sentiment analysis is presented.

Specifically, in focus on the Arabic syntax, grammar and morphology to create new sentences with the same labels or opposite labels as explained in Section 4.

The syntax of Arabic language is complex [37, 38] - as several matching cases are possible between words in the same sentence, while in addition, each word has several synonyms. Therefore, it is possible to generate tens of variants for an Arabic sentence while preserving its meaning. This task can be automated if the system is able to parse the statement and link it to lexical resources. Parsing is the process where each word in the text is labeled with its part of the speech tag (Verb, Object, Subject, etc.). However, parsing is not a simple process especially for Arabic language where the structure and order of the words are not specified. The Natural Language Processing Group at Stanford University has built an open-source parser [39]. Stanford Parser provides a set of natural language processing functions. Mainly, it was built for English, later on, many developers have carried out extensive work to improve the code and the grammatical rules to make it more comprehensive. As a result, this parser has been extended to include languages other than English, such as Chinese, German, Italian and Arabic. The parsing tool takes a text file as input and generates the base forms of words, normalizes and interprets dates, times, and numeric quantities. Finally, it analyzes the grammatical structure of the sentences. The output of the parsing process can be presented in several forms, such as phrase structures, trees, or dependencies.

For building the framework, initially the Stanford Arabic Parser was used to generate the parse trees of Arabic sentences. Afterwards, the augmentation rules generated were used on these trees, to generate several equivalent parse trees for the original sentences utilizing Arabic morphology, syntax, synonyms, and negation particles. These augmentation rules can be broadly divided into: (1) rules which alter or swap branches of the parse trees as per Arabic syntax and thus generate new sentences with the same labels (2) rules which generate new parse trees by utilizing the synonyms of words in these sentences, and also generate new sentences with the same original labels (3) rules which insert negation particles into the sentences and thus generate new sentences with opposite labels. It is worth mentioning here that the work in this paper addresses text augmentation for sentiment analysis. This means that the labels of the investigated sentences are either neutral, positive or negative. Applying the sets of rules described in (1) or (2) above will generate new sentences with the *same* labels as the input sentences. By comparison, applying the set of rules described in (3) as aforementioned, generates new sentences with *opposite* labels to the input sentences. Experiments proved the viability and effectiveness of the augmentation framework by running three experiments using three datasets. The size of the original datasets substantially increased and the generated sentences were of high quality.

The rest of this article is organized as follows: Section 2 briefly describes the related literature works. Section 3 explains the properties of Arabic language. Section 4 explains the design of the transformation rules which are the core of the augmentation framework. Section 5 describes the

implementation of the framework. Section 6 demonstrates the experiments which were carried out to assess the effectiveness of the proposed work. And finally section 7 summarizes the conclusion of this work.

## Related Work

This section describes related studies which have utilized Arabic WordNet as a component of frameworks. It also describes the related work which addresses data augmentation.

## Arabic WordNet

WordNet [40] is a large linguistic database, or hierarchical dictionary, which was initially developed for the English language. It has been very useful for the fields of computational linguistics and Natural Language Processing [41]. Because of its structure, the WordNet differs from other standard dictionaries, where it groups words based on their meanings. The English WordNet lexicon [42] is divided into syntactic categories that contain (nouns, verbs, adjectives, and adverbs). It should be noted here that function words are deleted. However, WordNet grouped synonyms using the meaning (thesaurus) rather than the form (dictionaries). It also represents words redundantly - where a given word may appear in noun, verb and adverb syntactic categories. The WordNet consists of four parts [40]: (1) lexicographers source files; (2) the tool to convert these files into the lexical database; (3) the lexical database; (4) software tools that are used to access the database.

WordNet has been very useful as it was used to build many Natural Language Processing applications, Information Retrieval, term expansion and document representations [43]. For example, Varelas et al. [44] compared the performance of using single ontology and different ontologies for the semantic similarity methods. Single ontology experiments were performed using the WordNet and it showed better performance in the results.

However, many efforts have been reported to adapt WordNet for other languages, such as WordNets for European languages [45], and French and Slavonian WordNets [46]. By comparison, Arabic WordNet [47] used the same development approach for word representation of Princeton WordNet to keep it compatible with other Word-Nets' structures. Arabic WordNet is a lexical database for Modern Standard Arabic, with two main linguistic categories (verbs and nouns). First, the important concepts that represent the core WordNet were extracted, then specific concepts for the Arabic language were developed along with other concepts that were manually translated to the most convenient synset from other languages. It was developed using MySQL and XML [47]. Finally, the Arabic WordNet ended up with 11,270 synsets (2538 verbs, 7961 nominal, 110 adverbs, and 661 adjectives) with 23,496 Arabic expressions. Table 1 presents detailed information about the statistical properties of Arabic WordNet.

Several researchers have targeted extending Arabic WordNet. For example, in the work reported in [48, 49], the authors automatically extracted named entities from Arabic Wikipedia. Subsequently, they attached these entities as instances to the synsets of Arabic WordNet and

finally created a link to their counterparts in English WordNet. Moreover, Badaro et. al [50] introduced an automatic method for expanding Arabic WordNet – where they formulated the problem as a link prediction problem.

Shoaib et al. [51] used the relationships in Arabic WordNet in order to build a model for semantic search in the Holy Quran. The proposed model improved searching and retrieving of the related verses from the Holy Quran without mentioning a specific keyword in the query. The model works in two stages. Namely, it identifies one sense of the query word using Word Sense Disambiguation, then it extracts out all the synonyms of the identified sense of the word. AlMaayah et al. [52] have also worked on the Holy Quran, where the researchers have built a model that extracts the synonyms and builds the Quranic Arabic WordNet. This net was built based on the Boundary Annotated Quran Corpus, lexicon resources, and traditional Arabic dictionaries. The final model was able to link the Holy Quran words that have the same meaning and generate sets of synsets using the vector space model. The Quranic Arabic WordNet has 6918 synsets from 8400 unique word senses. In other studies, the researches have tried to extract semantic relationships between words, and provide models to represent ontological relations for the Arabic content on the internet. These representations are useful to facilitate the analyses and processing of Arabic text. Al Zamil and Al-Radaideh [53] have used the semantic features that were extracted from the text along with syntactic patterns of relationships to provide models that are able to automate the process of ontological relations extraction. The extracted features are used to construct generalized rules which were used to build a classifier. The classifier presents each concept with its designated relationship label.

## Data Augmentation

Data augmentation is a technique that is used to increase the size of datasets and preserve the labels at the same time. It became popular with deep learning networks as they require training on huge datasets to secure high accuracies [54, 55, 56, 57]. Extending the size (number of samples) in a dataset, especially for under-represented classes, is mainly depended on generating perturbed replicas of the class samples. This technique has proved its success in image classification such as the work reported in [54, 58, 59]; 3D pose estimation as reported in [60]; speaker language identification as described in [61]; recognition of audio-visual effect [62]; and the classification of the environmental sound [63].

On the other hand, data augmentation is limited when dealing with textual data. This is due to the very difficult definition and standardization of specific rules or transformations that preserve the meaning of the produced textual data [19]. Basically, the main approach that works to increase the size of textual data, and preserves text meaning, is to use the synonyms of words, relying on lexical resources such as WordNet.

The works reported in [64, 65] have used a synonyms-based approach for augmenting textual data. As the synonyms are very limited, the proposed sentences are not very different and

numerous from the original texts. Therefore, Kobayashi [19] has proposed the contextual augmentation method, which is a state-of-the-art method to augment words, and produce more varied sentences. The author used words predicted by the bidirectional language model (LM) instead of using synonyms. The proposed approach was able to present a wide range of substitute words and it has been tested with two classifiers using recurrent or convolutional neural networks where it improves the overall performance. Georgios Rizos et al. [66] targeted extending a text used for hate speech detection relying on synonyms lists, wrapping the word token around the padded sequence, and finally applying class-based conditional recurrent neural language generation. The authors state that they achieved a 5.7% increase on Macro-F1 and a 30% in recall when extending the datasets using their three text extensions methods. The work reported in [67] has described a framework for augmenting tweets based on ConceptNet and Wikidata. The authors suggested two methods for improving the quality of tweets by first appending terms extracted from ConceptNet and Wikidata to the existing tweets but not increasing their numbers. Secondly, they generated new tweets by replacing words or terms in the original tweets with terms extracted from ConceptNet and Wikidata. This approach is close to the approaches which utilize synonyms. In a similar study, Kolomiyets et al. [68] replaced the headwords with a substitute word predicted from the Latent Words in the language model. The authors only used the top k score words as a substitute. Mueller and Thyagarajan [69] substituted random words in sentences with their synonyms to generate new sentences. Subsequently, they trained a siamese recurrent network to compute the similarity between sentences. Wang and Yang [65] employed word embedding to increase the size of the training data. Specifically, they replaced a given word with its nearest neighbor word vector. As it can be seen from the above literature, most of the existing augmentation techniques address image or audio data and less work addresses text augmentation. In this regard, it should be mentioned that no work addresses Arabic text augmentation. The current proposed framework is substantially different from text augmentation which relies on the replacement of words by their synonyms. On the other hand, it utilizes the rich syntax and grammar of the Arabic language in order to generate transformation rules, that are subsequently used to generate new sentences based on seed sentences.

## Arabic Language Properties

Arabic language is one of the Semitic languages. It consists of twenty-eight basic letters. Several Arabic letters change their shapes based on their location in the word. For example, the letter (س) has the shape (س) when it is located at the beginning of the word, the shape (س) when it is located at the middle of the word, (س) when it is located at the end of the word but connected to the previous letter, and (س) when it comes at the end of the word but disconnected from the previous letter. Arabic is an inflectional language that is written from right to left. The following three subsections provide background about Arabic language.



## Arabic Morphology

Morphology is the structure of words. The morphology of Arabic language is complex but systematic - where there are two ways to build a word in Arabic: derivation and agglutination. The derivation is a way of generating stems from a list of roots; based on three basic letters ( ف، ل، ع) for triliteral roots. For example, by using the root word “درس” that rhymes with “فعل” one can generate the following stems:

- Study “darasa دَرَسَ”
- Scholar “dAris دَارِس”
- Lesson “daros دَرَسَ”
- Teacher “mudaris مُدَرِّس”
- Schools “mdAris مَدَارِس”
- School “madrasop مَدْرَسَة”
- Study “mudArasop مُدَارَسَة”

The second way to build words in Arabic language is agglutination. In this way, the words are built by adding affixes to the word. These affixes could be prefixes at the beginning of words such as (ان، ت، تم، است)، infixes in the middle of the word (such as اِ), or suffixes at the end of the word such as (ة، اء، ان).

## Arabic Syntax

In Arabic scripts, the sentence has two types or categories (nominal and verbal). Each type has its own grammar and rules. The nominal sentence, in Arabic, consists of a subject (Almubtada) and predicate (Alkhabar). The normal order is that the subject is followed by the predicate but in certain cases, it is allowed to swap them (e.g. the sentence “أنت مجتهدٌ” which means “You are diligent” could be “مجتهدٌ أنت”). The subject in the nominal sentence can be Noun, Pronoun or Number while the predicate can be Singular Noun, Adverb, Preposition, Nominal sentence, or Verbal Sentence.

The verbal sentence in Arabic, like in many other languages, consists of Verb (V), Subject (S), and Object (O) without a specific order, which means that the order of verbal sentences could be: VSO, VOS, SVO or VOS. Additionally, in Arabic language diacritics, prefixes and suffixes are used to represent gender. Therefore, the absence of diacritics can create ambiguity and might change the meaning.

## Diacritics

One of the Arabic language features is the diacritics that are written above or underneath its letters. Diacritics are small vowel marks that represent three short vowels (a, i, u). They are used to regulate and control the letters and pronunciation. Therefore, diacritics have a huge effect on the text and its meaning, removing them may lead to morphological-lexical and morphological-syntactical ambiguities. For example, the word (nEm نعم) has the meaning 'Yes' if it was written

(naEom نَعْم), while it means 'graces' if it was written (niEm نِعَم). The basic diacritics of Arabic language are:

- Fatha: symbolized as an italic score on the top of the letter (-).
- Dma: symbolized as a small (و) letter on the top of the letter (-).
- Ksra: symbolized as an italic underscore on the bottom of the letter (-).
- Sokon: symbolized as a small circle on the top of the letter ( ).

## Transformation rules definition

As a first step, clear definitions of Arabic grammar rules were specified. These rules include specifications for nominal sentences, verbal sentences, questions, verbs, adjectives, pronouns, prepositions, conjunctions, and numbers. These defined grammar-based rules were represented using the Stanford Arabic parser tagset. Table 2 lists these tags in full details. Table 3, on the other hand, summarizes the core concepts of this research – it depicts, in the second column, grammar rules for valid sentences in Arabic. The third column of Table 3 lists equivalent grammar rules which were derived from the original rules listed in the second column. The importance of these rules is that sentences that respect the grammar rules listed in the second column could be mapped to new sentences which fulfill the grammar rules listed in column 3, and still have the same label for the classifiers. The following statements show example sentences from Arabic which respect grammar rules in Table 3 and show how these sentences are transformed into new sentences,  $\rightarrow$  means that the RHS of the rules are equivalent to the LHS:

**RULE 1:** DTNN +ADJ  $\rightarrow$  ADJ +DTNN

**Example:** Alrjl mHbwb (الرجل محبوب)  $\rightarrow$  mHbwb Alrjl (محبوب الرجل)

**Parse:** (ROOT (S (NP (DTNN الرجل)) (ADJP (JJ محبوب))))  $\rightarrow$  (ROOT (ADJP (JJ محبوب) (NP (DTNN الرجل))))

**RULE 2:** NN +ADJ  $\rightarrow$  ADJ +NN

**Example:** mAlk rA}E (مالك رائع)  $\rightarrow$  rA}E mAlk (رائع مالك)

**Parse:** (ROOT (S (NP (NNP مالك)) (ADJP (JJ رائع))))  $\rightarrow$  (ROOT (FRAG (NP (JJ رائع)) (NP (NNP مالك))))

**RULE 3:** DTNN +NN  $\rightarrow$  NN + DTNN

**Example:** Alrjl \$jAE (الرجل شجاع)  $\rightarrow$  \$jAE Alrjl (شجاع الرجل)

**Parse:** (ROOT (S (NP (DTNN الرجل)) (NP (NNP شجاع))))  $\rightarrow$  (ROOT (ADJP (JJ شجاع) (NP (DTNN الرجل))))

**RULE 4:** NN +NN  $\rightarrow$  NN + NN(swap)

**Example:** AHmd Swth rA}E (احمد صوته رائع)  $\rightarrow$  rA}E AHmd Swth (رائع احمد صوته)

**Parse:** (ROOT (S (NP (NNP احمد)) (NP (NN صوته) (JJ رائع))))  $\rightarrow$  (ROOT (FRAG (NP (JJ رائع) (NP (NNP احمد) (NNP صوته)))))

**RULE 5:** NN +DTNN  $\rightarrow$  NN + DTNN

- 352 **Example:** Ebd AlrHmn xlwq (عبد الرحمن خلق) → xlwq Ebd AlrHmn (خلق عبد الرحمن)
- 353 **Parse:** (ROOT (FRAG (NP (NNP عبد)) (NP (DTNNP الرحمن)) (NP (NNP خلق)))) → (ROOT (NP
- 354 (NNP خلق) (NNP عبد) (DTNNP الرحمن)))
- 355
- 356 **RULE 6:** DTNN + DTNN → DTNN + DTNN (swap)
- 357 **Example:** Ebd AlrHmn AlrHym (عبد الرحمن الرحيم) → AlrHym Ebd AlrHmn (الرحيم عبد الرحمن)
- 358 **Parse:** (ROOT (FRAG (NP (NNP عبد)) (NP (DTNNP الرحمن)) (NP (DTNNP الرحيم)))) → (ROOT
- 359 (S (NP (DTNNP الرحيم)) (NP (NNP عبد) (DTNNP الرحمن))))
- 360
- 361 **RULE 7:** ADJ + ADJ → DTNN + DTNN (sawap)
- 362 **Example:** AlftAp Aljmylp mjthdp (الفتاة الجميلة مجتهدة) → AlftAp mjthdp Aljmylp (المجتهدة الجميلة)
- 363 **Parse:** (ROOT (NP (DTNN الفتاة) (DTJJ الجميلة) (DTJJ مجتهدة))) → (ROOT (NP (DTNN الفتاة) (DTJJ
- 364 المجتهدة) (DTJJ الجميلة)))
- 365
- 366 **RULE 8:** PP + (NN+DTNN) → place them in the beginning and reverse the sentence
- 367 **Example:** Ebr Alm\$rf En \$kr AlfSl (عبر المشرف عن شكر الفصل) → En \$kr AlfSl Ebr Alm\$rf (عن
- 368 شكر الفصل عبر المشرف)
- 369 **Parse:** (ROOT (S (VP (VBD عبر) (NP (DTNN المشرف)) (PP (IN عن) (NP (NN شكر) (NP (DTNN
- 370 الفصل)))))) → (ROOT (S (PP (IN عن) (NP (NN شكر) (NP (DTNN الفصل)))) (NP (NN عبر) (NP
- 371 (DTNN المشرف))))))
- 372
- 373 **RULE 9:** PP + DTNN → place them in the beginning and reverse the sentences
- 374 **Example:** bAsm yqdm \$y}A mn AlfKAhAt (باسم يقدم شيئاً من الفكاهات) → mn AlfKAhAt bAsm yqdm
- 375 \$y}A (من الفكاهات باسم يقدم شيئاً)
- 376 **Parse:** ROOT (S (NP (NNP باسم)) (VP (VBP يقدم) (NP (NP (NN شيئاً)) (PP (IN من) (NP (DTNNS
- 377 الفكاهات)))))) → (ROOT (S (PP (IN من) (NP (NN الفكاهات) (NP (NNP باسم)))) (VP (VBP يقدم) (NP
- 378 (NN شيئاً))))))
- 379
- 380 **RULE 10:** PP + (Special character VB | NN) → place them in the beginnig and reverse the
- 381 sentence
- 382 **Example:** tSAdq mE Al\*}Ab EIY >n ykwn f>sk mstEdA (تصادق مع الذئاب على أن يكون فأسك مستعداً
- 383 → Al\*}Ab EIY >n ykwn tSAdq mE Al\*}Ab f>sk mstEdA (على أن يكون تصادق مع الذئاب فأسك مستعداً)
- 384 **Parse:** (ROOT (S (VP (VBP تصادق) (NP (NN مع) (NP (DTNN الذئاب)) (PP (IN على) (NP (DTNN
- 385 أن)))) (S (VP (VBP يكون) (NP (NNP فأسك) (ADJP (JJ مستعداً)))))) → (ROOT (S (PP (IN على) (NP
- 386 (DTNN أن)) (VP (VBP يكون) (S (VP (VBP تصادق) (NP (NN مع) (NP (NNP الذئاب) (NNP فأسك))
- 387 (ADJP (JJ مستعداً))))))
- 388
- 389 **RULE 11:** Wh-prounoun at the end of the sentences → Move it to the beginning
- 390 **Example:** njH Al\*y \*hb AIY Almdrsp (نجح الذي ذهب الى المدرسة) → Al\*y \*hb AIY Almdrsp njH
- 391 (الذي ذهب الى المدرسة نجح)
- 392 **Parse:** (ROOT (S (VP (VBD نجح) (SBAR (WHNP (WP الذي)) (S (VP (VBD ذهب) (PP (IN الى) (NP
- 393 (DTNN المدرسة)))))) → (ROOT (S (SBAR (WHNP (WP الذي)) (S (VP (VBD ذهب) (PP (IN الى) (NP
- 394 (NP (DTNN المدرسة)))))) (VP (VBD نجح))))
- 395
- 396 **RULE 12:** Special adverb + (NN | VB | (special character VB | NN)) → (NN | VB | (special
- 397 character VB | NN)) + Special adverb

- 398 **Example:** AlEfw End Almqdrp (العفو عند المقدرة) → End Almqdrp AlEfw (عند المقدرة العفو)
- 399 **Parse:** (ROOT (NP (NP (DTNN العفو)) (NP (NN عند) (NP (DTNN المقدرة))))) → (ROOT (NP (NN
- 400 عند) (NP (NP (DTNN المقدرة)) (NP (DTNN العفو)))))
- 401 **Example:** frH Alwld bxbr AlrHlp qbl >n y\*hb (فرح الولد بخبر الرحلة قبل أن يذهب) → qbl >n y\*hb frH
- 402 Alwld bxbr AlrHlp (الرحلة بخبر الرحلة) قبل أن يذهب فرح الولد بخبر الرحلة
- 403 **Parse:** (ROOT (S (NP (NP (NN فرح) (NP (DTNN الولد)) (NP (NP (NN بخبر) (NP (DTNN الرحلة))
- 404 (NP (NN قبل) (NP (DTNN أن))))) (VP (VBP يذهب))) → (ROOT (S (NP (NN قبل) (NP (DTNN أن))
- 405 (VP (VBP يذهب) (NP (NN فرح) (NP (DTNN الولد)) (NP (NN بخبر) (NP (DTNN الرحلة)))))
- 406
- 407 **RULE 13:** Pronoun + (NN | VB | ADJ) → (NN | VB | ADJ) + Pronoun
- 408 **Example:** hy tjd AlxyATp (هي تجيد الخياطة) → tjd hy AlxyATp (تجيد هي الخياطة)
- 409 **Parse:** (ROOT (S (NP (PRP هي)) (VP (VBP تجيد) (NP (DTNN الخياطة))))) → (ROOT (S (VP (VBP
- 410 تجيد) (NP (PRP هي)) (NP (DTNN الخياطة)))))
- 411 **Example:** Ant rjl krym (أنت رجل كريم) → rjl krym Ant (رجل انت كريم)
- 412 **Parse:** (ROOT (S (NP (PRP انت)) (NP (NP (NN رجل) (NP (NNP كريم))))) → (ROOT (NP (NP
- 413 (NN رجل) (NP (PRP انت)) (NP (NNP كريم)))))
- 414 **Example:** hy Al>jml (هي الأجمل) → hy Al>jml (هي الأجمل)
- 415 **Parse:** (ROOT (S (NP (PRP هي)) (NP (NNP الاجمل))))) → (ROOT (S (VP (VBP الاجمل) (NP (PRP
- 416 هي)))))
- 417
- 418 **RULE 14:** (NN|DTNN) + VB → VB+(NN|DTNN)
- 419 **Example:** Alwld y>kl qlyla (الولد يأكل قليلا) → y>kl Alwld qlyla (يأكل الولد قليلا)
- 420 **Parse:** (ROOT (S (NP (DTNN الولد)) (VP (VBP يأكل) (NP (NN قليلا))))) → (ROOT (S (VP (VBP
- 421 يأكل) (NP (DTNN الولد)) (NP (NN قليلا)))))
- 422 **Example:** bAsm yqdm \$y}A mn AlfkaAhAt (باسم يقدم شيئا من الفكاهات) → yqdm bAsm \$y}A mn
- 423 AlfkaAhAt (يقدم باسم شيئا من الفكاهات)
- 424 **Parse:** (ROOT (S (NP (NNP باسم)) (VP (VBP يقدم) (NP (NP (NN شيئا)) (PP (IN من) (NP (DTNNS
- 425 الفكاهات))))) → (ROOT (S (VP (VBP يقدم) (NP (NNP باسم)) (NP (NP (NN شيئا)) (PP (IN من) (NP
- 426 (DTNNS الفكاهات)))))
- 427
- 428 **RULE 15:** NN+ (Special-character + VB) → (special-character + vb) + NN
- 429 **Example:** Alwld <n ydrs ynjH (الولد إن يدرس ينجح) → <n ydrs Alwld ynjH (إن يدرس الولد ينجح)
- 430 **Parse:** (ROOT (S (NP (DTNN الولد) (DTJJ إن)) (VP (VBP يدرس) (S (VP (VBP ينجح))))) → (ROOT
- 431 (S (VP (VBD إن) (S (VP (VBP يدرس) (NP (DTNN الولد) (S (VP (VBP ينجح)))))
- 432 **Example:** bAsm lw ynAm mbkrA lA ytEb (باسم لو ينام مبكرا لا يتعب) → lw ynAm bAsm mbkrA lA
- 433 ytEb 0 (لوينام باسم مبكرا لا يتعب)
- 434 **Parse:** (ROOT (S (NP (NNP باسم)) (SBAR (IN لو) (S (VP (VBP ينام) (ADJP (JJ مبكرا))))) (PRT (RP
- 435 لا)) (VP (VBP يتعب))) → (ROOT (S (SBAR (IN لو) (S (VP (VBP ينام) (NP (NNP باسم)) (ADJP (JJ
- 436 لا)) (VP (PRT (RP لا)) (VBP يتعب)))))
- 437
- 438 **RULE 16:** VB+(NN|DTNN) → (NN+DTNN) + VB
- 439 **Example:** wqE Alwld EIY Al>rD (وقع الولد على الأرض) → Alwld wqE EIY Al>rD (الولد وقع على
- 440 الأرض)
- 441 **Parse:** (ROOT (S (VP (VBD وقع) (NP (DTNN الولد)) (PP (IN على) (NP (DTNN الأرض))))) →
- 442 (ROOT (S (NP (DTNN الولد)) (VP (VBD وقع) (PP (IN على) (NP (DTNN الأرض)))))
- 443

- 444 **RULE 17:** VB+(Special-character+(NN|DTNN)) → Special-character+(NN|DTNN)) +VB  
 445 **Example:** Elmt >n Alwfa' Sfp EZymp (علمت أن الوفاء صفة عظيمة) → >n Alwfa' Elmt Sfp EZymp  
 446 (أن الوفاء علمت صفة عظيمة)  
 447 **Parse:** (ROOT (S (VP (VBD علمت) (NP (NN أن) (NP (DTNN الوفاء)))) (NP (NN صفة) (JJ عظيمة))))  
 448 → (ROOT (S (NP (NN أن) (NP (DTNN الوفاء))) (VP (VBD علمت) (NP (NN صفة) (JJ عظيمة))))))  
 449  
 450 **RULE 18:** (Special-character +VB) + (Special-character+(NN|DTNN)) → (Special-character +  
 451 (NN|DTNN)) + (Special-character +VB)  
 452 **Example:** ln >Elm >n Alwfa' Sfp EZymp (لن أعلم أن الوفاء صفة عظيمة) → >n Alwfa' ln >Elm Sfp  
 453 EZymp (أن الوفاء لن أعلم صفة عظيمة)  
 454 **Parse:** (ROOT (S (VP (PRT (RP لن)) (VBP أعلم) (NP (NN أن) (NP (DTNN الوفاء)))) (NP (NN صفة)  
 455 (JJ عظيمة)))) → (ROOT (S (NP (NN أن) (NP (DTNN الوفاء))) (VP (PRT (RP لن)) (VBP أعلم) (NP  
 456 (NN صفة) (JJ عظيمة))))))  
 457  
 458 **RULE 19:** (Special-character +VB) +(NN|DTNN) → (NN|DTNN) + (Special-character +VB)  
 459 **Example:** ln >Elm Alwld n\$yT (لن أعلم الولد نشيط) → Alwld ln >Elm n\$yT (الولد لن أعلم نشيط)  
 460 **Parse:** (ROOT (S (VP (PRT (RP لن)) (VBP أعلم) (NP (DTNN الولد)) (ADJP (JJ نشيط)))) → (ROOT  
 461 (S (NP (DTNN الولد)) (VP (PRT (RP لن)) (VBP أعلم) (ADJP (JJ نشيط))))))  
 462  
 463 **RULE 20:** (Special-character+(NN|DTNN)) +VB → VB+(Special-character+(NN|DTNN))  
 464 **Example:** >n Alwfa' yEml mEk (ان الوفاء يعمل معك) → yEml >n Alwfa' mEk (يعمل أن الوفاء معك)  
 465 **Parse:** (ROOT (S (NP (NN أن) (NP (DTNN الوفاء))) (VP (VBP يعمل) (NP (NN معك)))) → (ROOT  
 466 (S (VP (VBP يعمل) (NP (NN أن) (NP (DTNN الوفاء) (DTJJ معك))))))  
 467  
 468 **RULE 21:** (Special-character+(NN|DTNN)) + (Special-character +VB) → (Special-character  
 469 +VB) +  
 470 (Special-character+(NN|DTNN))  
 471 **Example:** >n Alwfa' lA yHlw mEk (ان الوفاء لا يحلو معك) → lA yHlw >n Alwfa' mEk (لا يحلو أن الوفاء  
 472 معك)  
 473 **Parse:** (ROOT (S (NP (NN أن) (NP (DTNN الوفاء))) (VP (PRT (RP لا)) (VBP يحلو) (NP (NN معك))))  
 474 → (ROOT (S (VP (PRT (RP لا)) (VBP يحلو) (NP (NN أن) (NP (DTNN الوفاء) (DTJJ معك))))))  
 475  
 476 **RULE 22:** CD+(NN|DTNN|VB) → CD+(NN|DTNN|VB)  
 477 **Example:** njH Talb fy Altwjyhy (نجح 15 طالب في التوجيهي) → fy Altwjyhy njH Talb (في التوجيهي  
 478 15 طالب)  
 479 **Parse:** (ROOT (S (VP (VBD نجح) (NP (CD 15) (NP (NN طالب)))) (PP (IN في) (NP (ADJP (DTJJ  
 480 (التوجيهي)))))) → (ROOT (S (PP (IN في) (NP (ADJP (DTJJ (التوجيهي)))) (VP (VBD نجح) (NP (CD  
 481 15) (NP (NN طالب))))))  
 482  
 483 **RULE 23:** WH-Adverb +(NN|VB|DTNN|(Special-character+(NN|DTNN)) | (Special-character  
 484 +VB)) → (NN|VB|DTNN|(Special-character+(NN|DTNN)) | (Special-character +VB))+ WH-  
 485 Adverb  
 486 **Example:** kyf kl AlnAs y>klwn (كيف كل الناس يأكلون) → kl AlnAs kyf y>klwn (كل الناس كيف يأكلون)  
 487 **Parse:** (ROOT (SBARQ (WHADVP (WRB كيف)) (S (NP (NOUN\_QUANT كل) (NP (DTNN  
 488 (الناس))) (S (VP (VBP يأكلون)))))) → (ROOT (S (NP (NOUN\_QUANT كل)) (VP (NP (DTNN الناس))  
 489 (SBAR (WHADVP (WRB كيف)) (S (VP (VBP يأكلون))))))

Extensive experiments, showed that the Arabic Stanford parsing is not very accurate especially for the adverbs and negation words. This will adversely affect the system by generating wrong synonyms for the sentences. Therefore, there existed the need for declaring our own list of some adverbs, negation and special words because Stanford Parser does not assign the proper labels, as expected. These lists are presented in Table 4.

## Description of Framework

The general framework is illustrated in Figure 1. In the first step, a sentence with its label is passed to the system and it is converted to its canonical form (i.e. its parse tree). Secondly, multiple equivalent sentences (parse trees) are generated from the input sentence by replacing words with their synonyms. The synonyms are generated using Arabic WordNet. Thirdly, multiple variants of the sentences (parse trees), which were generated in step 2, are produced based on the transformation grammatical rules described in Table 3. The sentences generated in step 2 and step 3 all have the same label as the original input sentence. The Negation module is optionally called if we want to infuse negation particle into the generated sentences, and thus substantially increasing the number of generated sentences. The generated sentences from the Negation module have opposite labels to the input sentence and its variants. The following subsections describe, in detail, each developed module.

### Generate synonyms using Arabic WordNet

The Arabic WordNet browser is free and publicly available. It uses a locally-stored database of Arabic data in XML format - where words of the same meaning are linked through pre-defined lexical relations. Furthermore, the interface is modeled on the European language WordNet interface; hence, it contains the basic components with additional Arabic components. However, the performance of the Arabic WordNet is not satisfactory when compared with other WordNets. For example, the Arabic WordNet contains only 9.7% of the Arabic lexicon, while the English WordNet covers 67.5% of the English lexicon. Also, the Arabic WordNet synsets are linked only through hyponymy, synonymy, and equivalence; correspondingly seven semantic relations are used in the English WordNet. However, since the main goal is generating the synonyms of the words, the limitation of the Arabic WordNet did not substantially affect the work. Also, to avoid the noise caused by diacritics, only the first top five synsets in each synonyms list were considered. Table 5 shows the first eight synsets for the Arabic word “Man – رجل”. As it can be seen from Table 5, the further we go deeper in generating synonyms, the higher the chance of generating wrong synonyms. The last two entries in Table 5 correspond to “leg and foot” and not “Man”.

### Apply transformation rules to generate equivalent sentences

Employing the synonyms and the transformation rules, enables us to generate a huge number of sentences that are equivalent, in meaning and label, to the original input sentence. Every extracted synonym, using Arabic WordNet, creates a new sentence from the input sentence. Subsequently, these sentences are processed by the transformation module which selectively applies the proper transformation rules and generates even more sentences with the same meaning and label to the original sentence. Meaning, here, is defined in the loose sense of being suitable for sentiment analysis and is not from a linguistics perspective. From a linguistic perspective, synonymous sentences represent close meanings but not exactly the same. As an example, one can generate 47 sentences from the simple verbal sentence (“أكل الولد التفاحة”) (The boy ate the apple) using only the synonyms and transformation rules (i.e. without using the negation module which would generate even more sentences). Table 6 shows a sample of 9 sentences generated from the example sentence (The boy ate the apple).

### Generate Parse Trees

This module is responsible for generating parse trees for the original input sentence; and the generated sentences using Stanford Arabic parser tagset. With parse trees, it becomes easier to apply the suitable transformation rules to a given sentence, and it also facilitates the infusion of negation particles into sentences as described in the next section. Part (A) of Figure 2 depicts the parse tree of the sentence (أكل الولد التفاحة) (ate the boy the apple), while part (B) of Figure 2 shows the parse tree of the sentence (الولد أكل التفاحة) (the boy ate the apple). These two parse trees are equivalent.

### Negation

Negating a sentence in Arabic means inserting one of the negation particles used in Arabic into an affirmative sentence. Every negation particle, in Arabic, has its own rules in terms of the type of verbs or nouns it affects and in terms of the position in the sentence in which it is inserted. Negating a sentence will result in a new sentence that has an opposite meaning to the original input sentence. The label of the input sentence is also flipped from positive to negative. In addressing the negation problem, we adopted the Negation-aware Framework presented by Duwairi and Alshboul [70], where the authors explore the effects of Arabic morphology on sentiment analysis. The study focused on five negations particles (لم، لن، لا، ما، ليس) that have been grouped into two categories based on their effect on the word as shown in Table 7.

After defining the negation rules, the system is able to negate a set of sentences and generate all possible variations of these sentences as a result of inserting negation particles regardless if the sentences are nominal or verbal sentences. Table 8 shows an example of the output generated after applying negation to the positive verbal sentence (أعجب الولد الطعام) which means (The boy likes the food). As it can be seen from Table 8, this one sentence generates 10 sentences with

opposite labels (i.e. the input sentence shows positive sentiment towards food while the 10 generated sentences convey negative sentiments towards food).

## Evaluation

The following subsections describe, thoroughly, three experiments that were designed to test the accuracy of the proposed augmentation framework. Firstly, an assessment for the impact of the proposed framework on sentiment analysis was made. Secondly, we tested the correctness of each transformation rule. Finally, the accuracy of the Negation module was tested and formulated.

### Experiment1: classification of sentiment towards products

The aim of this experiment is to classify product reviews into positive, negative or neutral reviews. The focus of this experiment is not the classifier, but to assess the resulting accuracy of using the proposed framework when enlarging the size of the dataset. To perform the first experiment, we used a subset of a public dataset of product reviews [71] which contains 300 reviews written in Arabic collected from souq.com. The data was annotated with three labels (1: positive 0: Neutral, -1: negative). In this experiment, and before performing any changes on the original data, the data was tested using several supervised classifiers (Naive Bayes, K-nearest neighbor, and Support vector machine). The data was divided into 70% for training and 30% for testing. All the classifiers used word embedding that is generated using AraVec with a dimension equals to 300 [72]. After the training process for each classifier, the testing phase for each classifier's performance and ability to classify the testing data was performed. Accuracy was used to assess the performance of each classifier. Accuracy is calculated by dividing the number of correctly classified reviews by the number of all reviews. The reported accuracy was equal to 54.18% using the SVM classifier, 49.99% using the Naïve Bayes classifier, and 52.17% using the K-nearest neighbor classifier. Next, the data was fed into the augmentation tool where the size of the data was increased by almost 10 times. The generated dataset was tested using the same classifiers. In comparison with the previous results, the accuracy was increased by 42% on average. In details, the accuracy rates obtained by each classifier, using the augmented dataset, were 97% using the SVM, 87% using the NB and 91.66% using the K-nearest neighbor as illustrated in Figure 3. This improvement was expected - as increasing the dataset size will subsequently improve the training process which leads to improving the overall performance of the classifier.

### Experiment 2: Testing the efficiency of each transformation rule

The aim of this experiment was to test the accuracy of each transformation rule independently. To achieve this goal, it was preferable to design a small artificial dataset, which consists of 40 statements with positive sentiment, 32 statements with negative sentiment, and 27 neutral statements. A total of 99 sentences were carefully designed to align with the 23 transformation



rules. Each sentence was processed by the augmentation tool, and thus several sentences were generated for each input sentence. The generated sentences were manually inspected to test their validity. Rule accuracy is a measure that evaluates the ability of a given rule to generate correct and meaningful sentences. Rule accuracy is calculated by dividing the number of correct sentences generated by a given rule by the number of all sentences generated by that rule. "A correct sentence" means a grammatically correct and meaningful sentence. Table 9 shows the accuracy that was obtained for each rule. As can be seen from the table, all of the rules secured high accuracies. This means that the rules are capable of generating correct sentences. When examining the sources of error, we discovered that it was caused by improper synonymous words generated by the Arabic WordNet. It is important to note here that Arabic WordNet covers only 9.7% of the Arabic lexicon or vocabulary.

### Experiment 3: The efficiency of negation rules

The goal of the third experiment is to assess the capability of the Negation module in order to generate correct sentences. A small artificial dataset which consists of 26 positive sentences and 24 negative sentences was created for this purpose. It should be mentioned here that the Negation module is responsible for inserting proper negation particles into the input sentences. Negation flips the polarity of the input sentence. This means that positive sentences will become negative and vice versa. All the resulted sentences from the Negation module are correct with their respective labels properly flipped.

### Conclusion

In this study, a novel data augmentation framework for Arabic textual datasets for sentiment analysis was presented. In total, 23 transformation rules were designed to generate new sentences from the input ones. These rules were designed after carefully inspecting Arabic morphology and syntax. To increase the number of generated sentences for every rule, Arabic WordNet was used to swap the words with their respective synonyms. These rules preserve the labels of the input sentences. This means that if the input sentence has a positive label then the generated sentences also have positive labels. By the same token, if the label of the input sentence is negative, the labels of the generated sentences are also negative. The same is true for the neutral label. A Negation module was also designed to insert negation particles into Arabic sentences. This module inverts or flips the labels of the generated sentences, as this is the effect of negation particles on the polarity of statements. Experimentally, we tested the proposed framework by conducting three experiments. The first experiment has demonstrated the effect of increasing the dataset size, using the augmentation tool, on classification. As expected, the accuracy improved in all the classifiers. This indicates that the quality of the generated sentences was high. The second experiment was designed to test the accuracy of each transformation rule. An artificial dataset was designed for this purpose. All rules scored extremely high accuracies. The third and last experiment used an artificial dataset to assess the quality of the generated sentences from the

Negation module. The experiment reveals that all generated sentences were correct with proper associated labels.

# Acknowledgement

This work was supported by Jordan University of Science and Technology, Jordan. Grant No. (20160150).

# References

- [1] Stefan Weninger, Geoffrey Khan, Michael P. Streck, Janet C. E. Watson (ed). Semitic languages: an international handbook, Walter de Gruyter gmbh & Co. Berlin/Boston, 2011.
- [2] Ibrahim Al-Huri. Arabic Language: Historic and Sociolinguistic Characteristics. 2015. English Literature and Language Review, Vol 1, Issue 4, pages 28-36.
- [3] Summary by language size. Ethnologue. <https://www.ethnologue.com/statistics/summary-language-size-19>. Last retrieved 30-10-2020.
- [4] Documentation for ISO 639 identifier: ara. <https://iso639-3.sil.org/code/ara>. Last retrieved 30-10-2020.
- [5] Korayem Mohammed, David Crandall, and Muhammad Abdul-Mageed. Subjectivity and sentiment analysis of Arabic: A survey. International conference on advanced machine learning technologies and applications. Springer, Berlin, Heidelberg, 2012.
- [6] Mona Diab, Musa Alkhalifa, Sabri Elkateb, Christiane Fellbaum, Aous Mansouri, Martha Palmer. Semeval-2007 task 18: Arabic semantic labeling. Proceedings of the Fourth International Workshop on Semantic Evaluations (semeval-2007). 2007.
- [7] Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. Automatic tagging of Arabic text: From raw text to base phrase chunks. Proceedings of HLT-NAACL 2004: Short papers. Association for Computational Linguistics, 2004.
- [8] Green, Spence, and Christopher D. Manning. Better Arabic parsing: Baselines, evaluations, and analysis. Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010.
- [9] Stanford Arabic parser tagset | Sketch Engine, [sketchengine.co.uk](http://sketchengine.co.uk), 2018. [Online]. Available: <https://www.sketchengine.co.uk/stanford-arabic-parser-tagset/>. Last retrieved 30-10-2020.
- [10] Sarah Al-Humoud, Mawaheb Altuwaijri, Tarfa AlBuhairi, Wejdan Alohaideb. Survey on Arabic sentiment analysis in Twitter. International Science Index 9.1 (2015): 364-368.
- [11] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, Samir Belfkih. 2020. ASA: A framework for Arabic sentiment analysis. Journal of Information Science, vol. 46, Issue 4, pages 544-559, <https://doi.org/10.1177/0165551519849516>.
- [12] Abubakr H. Ombabi, Wael Ouarda, Adel M. Alimi. 2020. Deep learning CNN-LSTM framework for Arabic sentiment analysis using textual information shared in social networks. Social Network Analysis and Mining, Volume 10, issue 53, (2020). <https://doi.org/10.1007/s13278-020-00668-1>
- [13] Rehab Duwairi, Mosab Alfaqeh, Sentiment Analysis for Arabizi Text. The 7th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 5-7 April, 2016.

- [14] Ali Ahmed, Stephan Vogel, and Steve Renals. Speech recognition challenge in the wild: Arabic MGB-3. 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017.
- [15] Eiman Alsharhan, and Allan Ramsay. Improved Arabic speech recognition system through the automatic generation of fine-grained phonetic transcriptions. *Information Processing & Management*. Vol. 56, Issue 2, 2019, pages: 343 – 353.
- [16] Michel Galley, Spence Green, Daniel Cer, Pi-Chuan Chang, Christopher D. Manning (2009). Stanford University's Arabic-to-English Statistical Machine Translation System for the 2009 NIST Evaluation. The 2009 NIST Open Machine Translation Evaluation Meeting. Ottawa, Canada.
- [17] Ruqyya. Al-Ibrahim, Rehab Duwairi. Neural Machine Translation from Jordanian Dialect to Modern Standard Arabic, 11th International Conference on Information and Communication Systems, Irbid, 2020.
- [18] Bara Bataineh, Rehab Duwairi and Malak Abdullah, ArDep: An Arabic Lexicon for Detecting Depression, The 3rd International Conference on Advances in Artificial Intelligence (ICAAI 2019), Istanbul, Turkey, 26-28 Oct. 2019.
- [19] Sosuke Kobayashi. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*.
- [20] Oumaima Oueslati, Erik Cambria, Moez Ben HajHmida, Habib Ounelli, A review of sentiment analysis research in Arabic language, *Future Generation Computer Systems* 112 (2020), pp. 408–430.
- [21] Mahmoud Al-Ayyoub, Abed Allah Khamaiseha, Yaser Jararweha, Mohammed N. Al-Kabib, A comprehensive survey of Arabic sentiment analysis. *Information Processing and Management*, 56 (2019), pp. 320 – 342.
- [22] Yang Liu, Xiangji Huang, Aijun An, Xiaohui Yu, ARSA: a sentiment-aware model for predicting sales performance using blogs, in: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2007, pp. 607–614.
- [23] Andrea Ceron, Luigi Curini, Stefano M. Iacus, Using sentiment analysis to monitor electoral campaigns: Method matters - evidence from the United States and Italy, *Soc. Sci. Comput. Rev.* 33 (1) (2015) 3–20.
- [24] Monireh Ebrahimi, Amir Hossein Yazdavar, Amit Sheth. Challenges of Sentiment Analysis for Dynamic Events. *IEEE Intelligent Systems*, 2017, vol. 23, Issue 5.
- [25] Shanshan Gao, Jinxing Hao, Yu Fu, The application and comparison of web services for sentiment analysis in tourism, in: *Service Systems and Service Management (ICSSSM)*, 2015 12th International Conference on, IEEE, 2015, pp. 1– 6.
- [26] Erik Cambria, Marco Grassi, Amir Hussain, Catherine Havasi, Sentic computing for social media marketing, *Multimedia Tools and Applications*, 59 (2) (2012) 557–577.
- [27] Rehab Duwairi, Nizar Ahmed, Saleh Al-Rifai. Detecting sentiment embedded in Arabic social media – A lexicon-based approach. *Journal of Intelligent & Fuzzy Systems*, 2015, 29(1), pp. 107–117.
- [28] Rehab Duwairi, & Mahmoud El-Orfali. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *Journal of Information Science*, 2014, 40(4), pp. 501–513.

- [29] Siaw Ling Lo, Erik Cambria, Raymond Chiong, David Cornforth. Multilingual sentiment analysis: from formal to informal and scarce resource languages. *Artificial Intelligence Review*, (2017), 48:499–527.
- [30] David Vilares, Haiyun Peng, Ranjan Satapathy, Erik Cambria. BabelSenticNet: A Commonsense Reasoning Framework for Multilingual Sentiment Analysis. 2018 IEEE Symposium Series on Computational Intelligence (SSCI)
- [31] Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani. Cross-Lingual Sentiment Quantification. *IEEE Intelligent Systems*, 2020, vol. 35, Issue 3.
- [32] S. Zhai , and Z.M. Zhang. 2016. Semisupervised autoencoder for sentiment analysis. In Thirtieth AAAI Conference on Artificial Intelligence.
- [33] Tang, D., B. Qin, and T. Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 1014–1023.
- [34] Tang, D., B. Qin, and T. Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1422–1432.
- [35] Zhou, H., L. Chen, F. Shi, and D. Huang. 2015. Learning bilingual sentiment word embeddings for cross-language sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 430–440.
- [36] Le, Q., and T. Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196.
- [37] Abeer Salih Hadi. *Factors of Structural Ambiguity in English and Arabic: A Contrastive Study*. Psychology, 2015.
- [38] Daimi Kevin. Identifying syntactic ambiguities in single-parse Arabic sentence. *Computers and the Humanities*, 35, 3. 2001. Pages: 333-349.
- [39] The Stanford Natural Language Processing Group, Nlp.stanford.edu, 2018. [Online]. Available: <https://nlp.stanford.edu/software/lex-parser.shtml>.
- [40] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 235–244.
- [41] G. A. Miller, C. Fellbaum. 2007. WordNet then and now. *Language Resources and Evaluation*, 41, 209–214.
- [42] G. A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38, 39–41.
- [43] Christiane Fellbaum, Piek Vossen. *Connecting the universal to the specific: Towards the global grid*. International Workshop on Intercultural Collaboration. Springer, Berlin, Heidelberg, 2007.
- [44] Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides Petrakis, Evangelos Milios. Semantic similarity methods in WordNet and their application to information retrieval on the web. *Proceedings of the 7th annual ACM international workshop on Web information and data management*. ACM, 2005.
- [45] Piek Vossen. EuroWordNet: a multilingual database of autonomous and language specific wordnets connected via an Inter-Lingual-Index. *International Journal of Lexicography*, (2004), Vol.17 Issue 2, pages 161-173. <https://doi.org/10.1093/ijl/17.2.161>

- [46] Benoît Sagot, Darja Fišer. Extending WordNets by learning from multiple resources, HAL Archive, <https://hal.archives-ouvertes.fr/hal-00655785>, last accessed 12-Feb-2021.
- [47] S. Elkateb, W. Black, H. Rodri'guez, M. Alkhalifa, P. Vossen, A. Pease, C. Fellbaum. (2006). Building a WordNet for Arabic. In Proceedings of the fifth international conference on language resources and evaluation (LREC 2006).
- [48] Musa Alkhalifa, Horacio Rodríguez, Automatically Extending NE coverage of Arabic WordNet using Wikipedia. 3rd International Conference on Arabic Language Processing (CITALA'09), May 4-5, 2009, Rabat, Morocco.
- [49] Musa Alkhalifa and Horacio Rodríguez. Automatically Extending Named Entities Coverage of Arabic WordNet using Wikipedia. International Journal on Information and Communication Technologies, Vol. 3, No. 3, June 2010.
- [50] Gilbert Badaro and Hazem Hajj, Nizar Habash. A Link Prediction Approach for Accurately Mapping a Large-scale Arabic Lexical Resource to English WordNet. ACM Trans. Asian Low-Resource Language Information Processing, Vol. 19, No. 6, Article 80. pp. 1-38, 2020.
- [51] Muhammad Shoaib, M. Nadeem Yasin, Hikmat Khan, Muhammad Saeed, Malik Khiyal. 2009. Relational WordNet model for semantic search in Holy Quran. 2009 IEEE 5th International Conference on Emerging Technologies.
- [52] Manal Almaayah, Majdi Sawalha, and Mohammad Abushariah. Towards an automatic extraction of synonyms for Quranic Arabic WordNet. International Journal of Speech Technology 19.2 (2016): 177-189.
- [53] Mohammed Al Zamil, Qasem Al-Radaideh. Automatic extraction of ontological relations from Arabic text. Journal of King Saud University-Computer and Information Sciences 26.4 (2014): 462-472.
- [54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. 1097–1105.
- [55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [56] Navdeep Jaitly, Geoffrey E. Hinton. Vocal tract length perturbation (VTLP) improves speech recognition. Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language. Vol. 117. 2013.
- [57] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Sanjeev Khudanpur. Audio augmentation for speech recognition. Sixteenth Annual Conference of the International Speech Communication Association. 2015.
- [58] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. 2017. A bayesian data augmentation approach for learning deep models. In Advances in Neural Information Processing Systems. 2797–2806.
- [59] Suhib. Irsheidat, Rehab Duwairi, Brain Tumor Detection Using Artificial Convolutional Neural Networks. 11th International Conference on Information and Communication Systems, Irbid, 2020.
- [60] Grégory Rogez and Cordelia Schmid. 2016. Mocap-guided data augmentation for 3d pose estimation in the wild. In Advances in Neural Information Processing Systems. 3108–3116.

- [61] Gil Keren, Jun Deng, Jouni Pohjalainen, and Björn W Schuller. 2016. Convolutional Neural Networks with Data Augmentation for Classifying Speakers' Native Language. In Proceedings of the 17th International Speech Communication Association INTERSPEECH. Pages 2393–2397.
- [62] Panagiotis Tzirakis, George Trigeorgis, Mihalis Nicolaou, Björn W. Schuller, and Stefanos Zafeiriou. 2017. End-to-end multimodal emotion recognition using deep neural networks. IEEE Journal of Selected Topics in Signal Processing 11, 8 (2017), 1301–1309.
- [63] Justin Salamon and Juan Pablo Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters 24, 3 (2017), 279–283.
- [64] Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level convolutional networks for text classification. Advances in Neural Information Processing Systems. 2015.
- [65] Wang, W. Y. and Yang, D. (2015). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using petpeeve tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2557– 2563.
- [66] Georgios Rizos, Konstantin Hemker, Björn Schuller, Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification, CIKM '19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, November 2019 Pages 991–1000, <https://doi.org/10.1145/3357384.3358040>.
- [67] Sima Sharifirad, Borna Jafarpour, Stan Matwin. 2018. Boosting Text Classification Performance on Sexist Tweets by Text Augmentation and Text Generation using a Combination of Knowledge Graphs. 2018. Proceedings of the Second Workshop on Abusive Language Online (ALW2), pages 107–114.
- [68] Oleksandr Kolomiyets, Steven Bethard, Marie-Francine Moens. Model-portability experiments for textual temporal analysis. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.
- [69] Mueller, J. and Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In AAAI, pages 2786–2792.
- [70] Rehab Duwairi, Mohammad A. Alshboul. Negation-aware Framework for Sentiment Analysis in Arabic Reviews, The 2nd International Workshop on Social Networks Analysis, Management and Security (SNAMS - 2015), August 2015, Rome, Italy.
- [71] H. ElSahar, S. El-Beltagy. (2015) Building Large Arabic Multi-domain Resources for Sentiment Analysis. In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2015. Lecture Notes in Computer Science, vol 9042. Springer, Cham. [https://doi.org/10.1007/978-3-319-18117-2\\_2](https://doi.org/10.1007/978-3-319-18117-2_2).
- [72] Abu Bakr Soliman, Kareem Eisa, and Samhaa R. El-Beltagy, “AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP”, in proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, UAE, 2017.

# **Table 1**(on next page)

Statistical properties of Arabic WordNet.

Statistical properties of Arabic WordNet.

	Unique strings	Synsets
Noun	13,330	7961
Verb	5595	2536
Named entities	1426	1155
Broken plurals	405	126
Total	20,756	11,778

**Table 1.** Statistical properties of Arabic WordNet.



## Table 2 (on next page)

Stanford Arabic parser tageset.

Stanford Arabic parser tageset.

1

Tag	Description	Tag	Description
ADJ	Adj	NNS	Noun, plural
CC	Coordinating conjunction	NOUN	Noun
CD	Cardinal number	PRP	Personal pronoun
DT	Determiner	PRP\$	Possessive pronoun
DTJJ	Adjective with the determiner "Al" (ال)	PUNC	Punctuation
DTJJR	Adjective, comparative with the determiner "Al" (ال)	RB	Adverb
DTNN	Noun, singular or mass with the determiner "Al" (ال)	RP	Particle
DTNNP	Proper noun, singular with the determiner "Al" (ال)	UH	Interjection
DTNNPS	Proper noun, plural with the determiner "Al" (ال)	VB	Verb, base form
DTNNS	Noun, plural with the determiner "Al" (ال)	VBD	Verb, past tense
IN	Preposition or subordinating conjunction	VBG	Verb, gerund or present participle
JJ	Adjective	VBN	Verb, past participle
JJR	Adjective, comparative	VBP	Verb, non-3rd person singular present
NN	Noun, singular or mass	VN	Verb, past participle
NNP	Proper noun, singular	WP	Wh-pronoun
NNPS	Proper noun, plural	WRB	Wh-adverb

2

**Table 2.** Stanford Arabic parser tagset.

3

### **Table 3**(on next page)

Transformation rules based on Arabic grammar.

Transformation rules based on Arabic grammar.

ID	Original rules	Equivalent Rules
1	DTNN +ADJ	ADJ+DTNN
2	NN+ADJ	ADJ+NN
3	DTNN+NN	NN+DTNN
4	NN+NN	NN + NN (swap)
5	NN+DTNN	could not be changed
6	DTNN+DTNN	DTNN+ DTNN (swap)
7	ADJ+ADJ	ADJ +ADJ (swap)
8	PP+ (NN+DTNN)	Place at the beginning and reverse the sentence.
9	PP+(DTNN)	Place at the beginning and reverse the sentence.
10	PP+ (special character VB   NN)	Place at the beginning and reverse the sentence.
11	Wh-prounoun+ end of the sentences	Place at the beginning of sentences
12	Special adverb + (NN   VB   (special character VB   NN))	(NN   VB   (special character VB   NN)) + Special adverb
13	Pronoun+ (NN  VB   ADJ)	(NN  VB   ADJ) + Pronoun
14	(NN DTNN) +VB	VB+(NN DTNN)
15	NN+(Special-character +VB)	(special-character) +NN
16	VB+(NN DTNN)	(NN+DTNN) +VB
17	VB+(Special-character+(NN DTNN))	(Special-character+(NN DTNN)) +VB
18	(Special-character +VB) + (Special-character+(NN DTNN))	(Special-character+(NN DTNN)) + (Special-character +VB)
19	(Special-character +VB) +(NN DTNN)	(NN DTNN) + (Special-character +VB)
20	Special-character+(NN DTNN) +VB	VB+(Special-character+(NN DTNN))
21	(Special-character+(NN DTNN)) + (Special-character +VB)	(Special-character +VB) + (Special-character+(NN DTNN))
22	CD+(NN DTNN VB)	could not be changed
23	WH-Adverb +(NN VB DTNN (Special-character+(NN DTNN))   (Special-character +VB))	(NN VB DTNN (Special-character + (NN DTNN))   )Special-character + VB)) + WH-Adverb

**Table 3.** Transformation rules based on Arabic grammar.

**Table 4**(on next page)

List of special adverbs, negation and special words

List of special adverbs, negation and special words

1

Word	Transliteration	Meaning	Word	Transliteration	Meaning
قبل	qbl	Before	إن	Ena	that
بعد	bEd	After	أن	Ana	that
فوق	fwq	Above	فإن	fA'n	Then
تحت	tHt	Under	لو	lw	If
أسفل	Asfl	Down	كي	ky	So that
أمام	A mAm	In front of	لكي	lky	in order to
وراء	wrA'	behind	قد	qd	may
أعلى	AEIY	Top	لكن	lkn	But
وسط	wsT	Center	لم	lm	did not
عند	End	At	ما	mA	What
خلف	xlf	behind	بعض	bED	Some
شمال	\$mAl	north	فقط	fqT	Just
جنوب	jnwb	South	ليت	lyt	wish
شرق	\$rq	east	لعل	IEI	Might
غرب	grb	West	ذي	*y	The
يمين	ymyn	right	ليس	lys	Not
يسار	ysAr	left	إذا	E *A	if
كل	KL	Each			

2

**Table 4.** List of special adverbs, negation and special words

3

# **Table 5**(on next page)

Examples of synsets for “Man - رجل” using Arabic WordNet.

Examples of synsets for “Man - رجل” using Arabic WordNet.

1

Arabic	Transliteration	Meaning in English
ذكر	*kr	Male
عاشق	EA\$g	Lover
حبيب	Hbyb	Lover
قرين	qryn	consort
محبوب	mHbwb	Lover
زوج	zwj	Husband
قدم	qadam	Foot
ساق	saAq	Leg

2

**Table 5.** Examples of synsets for “Man – رجل” using Arabic WordNet.

3



# **Table 6**(on next page)

Examples of some equivalent sentences generated from the statement (أكل الولد التفاحة)

Examples of some equivalent sentences generated from the statement (أكل الولد التفاحة)

1

أكل الولد التفاحة Original:		
WordNet Synonymous		Rules Possibilities
Subject: الولد	1	أكل الولد التفاحة
	2	الولد أكل التفاحة
	3	أكل التفاحة الولد
Subject: الشاب	4	أكل الشاب التفاحة
	5	الشاب أكل التفاحة
	6	أكل التفاحة الشاب
Subject: الفتى	7	أكل الفتى التفاحة
	8	الفتى أكل التفاحة
	9	أكل التفاحة الفتى

2

**Table 7** (on next page)

Negation particles and their effects in Arabic [48]

Negation particles and their effects in Arabic [48]

1

Negation Particle	BuckWalter Transliteration	Category	Effect
لم	lam	Group A	Affects the verb after the particle
لن	Lan	Group A	Affects the verb after the particle
لا	lA	Group A	Affects the verb after the particle
ما	mA	Group A	Affects the verb after the particle
ليس	laysa	Group B	Affects the following two nouns or affects the following verb.

2

**Table 7.** Negation particles and their effects in Arabic [48]

3

# **Table 8**(on next page)

Examples of negated Arabic verbal sentences.

Examples of negated Arabic verbal sentences.

1

Negation Particles	Generated Sentences
ما	ما أعجب الولد الطعام
	الطعام ما أعجب الولد
لم	لم يعجب الولد الطعام
	الطعام لم يعجب الولد
لن	لن يعجب الطعام الولد
	الطعام لن يعجب الولد
لا	لا يعجب الطعام الولد
	الطعام لا يعجب الولد
ليس	ليس يعجب الولد الطعام
	ليس الطعام يعجب الولد

2

**Table 8.** Examples of negated Arabic verbal sentences.

3

**Table 9**(on next page)

Accuracy rate per transformation rule

Accuracy rate per transformation rule

1

<b>Rule 1</b>	<b>Rule 2</b>	<b>Rule 3</b>	<b>Rule 4</b>	<b>Rule 5</b>
77.5	89.3	90.07	94.28	97.53
<b>Rule 6</b>	<b>Rule 7</b>	<b>Rule 8</b>	<b>Rule 9</b>	<b>Rule 10</b>
91.50	96.36	94.4	100	100
<b>Rule 11</b>	<b>Rule 12</b>	<b>Rule 13</b>	<b>Rule 14</b>	<b>Rule 15</b>
98.79	96.96	100	100	100
<b>Rule 16</b>	<b>Rule 17</b>	<b>Rule 18</b>	<b>Rule 19</b>	<b>Rule 20</b>
100	100	97.5	100	100
<b>Rule 21</b>	<b>Rule 22</b>			
100	100			

2

**Table 9:** Accuracy rate per transformation rule

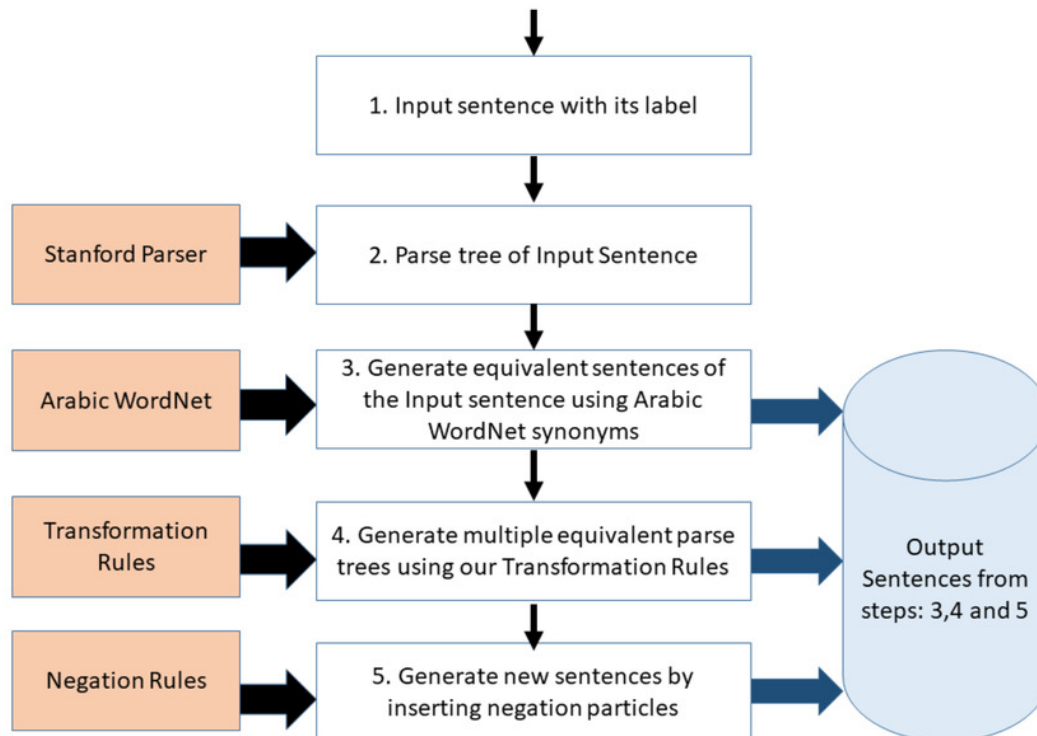
3



# Figure 1

Framework for the proposed text augmentation tool

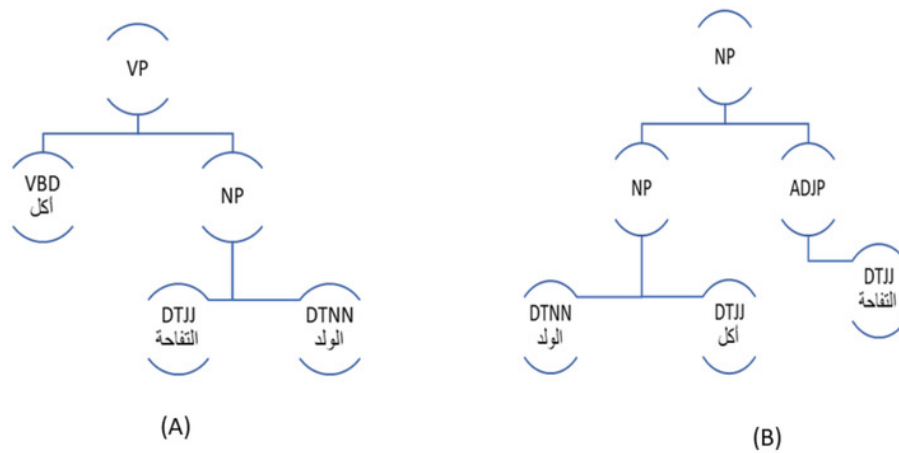
Framework for the proposed data augmentation tool



# Figure 2

equivalent parsing trees.

Examples of equivalent parsing trees.



# Figure 3

Accuracy rates using the original dataset and the augmented dataset

Accuracy rates using the original dataset and the augmented dataset .

