ISSN 2007-9737

Revisiting Arabic Morphology: A Machine-Learning-Based Approach to Stemming and Root Character Permutations with a Publicly Available Open-source Implementation

Iskander Akhmetov[1], Basem Ibrahim Malawi Al-Raba'a[2], Alexander Gelbukh[3,*], Rustam Mussabayev[1], Alexander Krassovitskiy[1]

[1] Institute of Information and Computational Technologies, Almaty, Kazakhstan

[2] The University of Jordan, Amman, Jordan

[3] Instituto Politecnico Nacional, Centro de Investigacion en Computacion, Mexico City, Mexico

gelbukh@cic.ipn.mx

Abstract. We present a novel method for learning morphological rules from a corpus for a major language with very rich and complicated morphology. Namely, we conducted experiments on Arabic stemming and root character permutations, focusing on their properties and semantic relations. First, we built stemmer and lemmatizer models using an only 0.2713/0.6347 accuracy score for the test/train sets. Second, we have explored the semantic relationship between word root character permutation variants. We have found that recombining the characters in a root may give rise to antonymy, synonymy, or other semantic relations. The paper is accompanied with an open-source implementation of the method freely available for research purposes.

Keywords. Stemming, lemmatization, Arabic, root, permutation.

1 Introduction

Arabic, as widely known, is a Semitic language with nonconcatenative and affixal morphology [12, 25, 42, 19, 30, 29].

Nonconcatenative or templatic morphology refers to word formation processes in which the consonantal roots are mapped onto a set of vocalic and consonantal templates, which results in the modification of meaning.

Words formed by combining roots with templates are of invariant shape and represent the core of the word structure to which affixes can be added. Table 1 contains some illustrative examples وصل.

Observe, first, that the consonants of the root must be in the same linear order, and second, that the root provides an abstract meaning that becomes modified by the verb form, i.e., the template onto which the root is mapped.

Affixal morphology, which generally adds grammatical information (e.g., person, number, and gender), comes into play after the internal structure of the root and template is formed. For instance, the suffix *-t* encoding the feminine marker can be

added to any verb form in the above table, as in *waṣala-t* (وَصَلَت) 'she arrived'.

As far as stemming is concerned, the verb forms like those in Table 1 can be considered basic stems since they lack any grammatical or inflectional affixes [24, 27].

Although Arabic morphology has been extensively studied, in this paper, we explore two of its understudied aspects — stemming and root character permutations — within the framework of computational linguistics. We show that there is room for the development of the Arabic language stemmers, and the permutation of the root characters can create semantic relations between the lexical items containing the permutated root, particularly synonymy and antonymy.

Our approaches consisted of the following steps; see Fig. 1 for reference:

1. **The Arabic stemmer**: We have trained a stemmer model using a previously developed technique [2].
2. **Antonyms/synonyms analysis**: We have classified the semantic relationships of root character permutation pairs, using the model obtained in step 3 and primarily focusing on antonyms and synonyms.
3. **Relationship classifier**: We have used the data from ConceptNet database [36] on the type of relationships between word pairs using a classifier model.

The contribution of our work is as follows:

1. The Arabic Stemmer model has beaten the ten well-known peers by a high margin .
2. Dataset with :
   - — The Arabic stemming results for 7,243 words in the test set by 10 different stemmers.
   - — Valid root character permutations for 51,476 words.
3. A Python module based on the current work called *cooc-ar-stemmer*[1]

[1] https://pypi.org/project/cooc-ar-stemmer/

The rest of the article is organized as follows: in the next section, we provide an overview of related works, describe the data and methodology we have used, and shed light on the experiment setup section.

Next, we present the results and discussion sections, followed by a conclusion.

2 Related Works

The topic of morphological analysis, including the tasks of lemmatization and stemming, has received considerable attention as researchers have continued to seek more efficient approaches to morphological analyses.

Tolegen et al. attempted to use a finite state transducer for the morphological analysis of the Kazakh language, an agglutinative language belonging to the Turkic family of languages [40]. Furthermore, Toleu et al. developed a language-independent approach for morphological disambiguation [41].

In Arabic derivational morphology, a root, consisting of two, three, or four consonants/radicals, basically carries a generic meaning modified based on the pattern with which the root combines.

For instance, the root ʕ-*l*-*m* (علم) expresses the generic meaning of knowing, but when this root is mapped onto distinct patterns, it yields slightly different, specified meanings as seen in ʕ*ilm* (عِلم) 'knowledge', ʕ*alima* (عَلِمَ) 'he knew', and ʕ*aalim* (عالِم) 'scholar (a man of knowledge)'.

Note that the linear order of the radicals in such examples is preserved. This type of derivational morphology (الاشتقاق الصغير = literally, the small derivation), which has been extensively examined in the literature, is a very productive process in Arabic as well as in other Semitic languages. Another type of derivation that has received very

Table 1. Illustrative examples of the Arabic root-and-pattern system.

Root	Abstract Meaning of the Root	Template	Meaning
w-ṣ-l (وصل)	'arriving/connecting'	Form 1: waṣala (وَصَلَ)	'he arrived'
		Form 2: waṣṣala (وَصَّلَ)	'he gave a ride to (someone)
		Form 3: waaṣala (واصَلَ)	'he continued'
		Form 4: ʔawṣala (أَوصَلَ)	'he led'
		Form 5: tawaṣṣala (تَوَصَّلَ)	'he arrived at'
		Form 6: tawaaṣala (تواصَلَ)	'he corresponded'

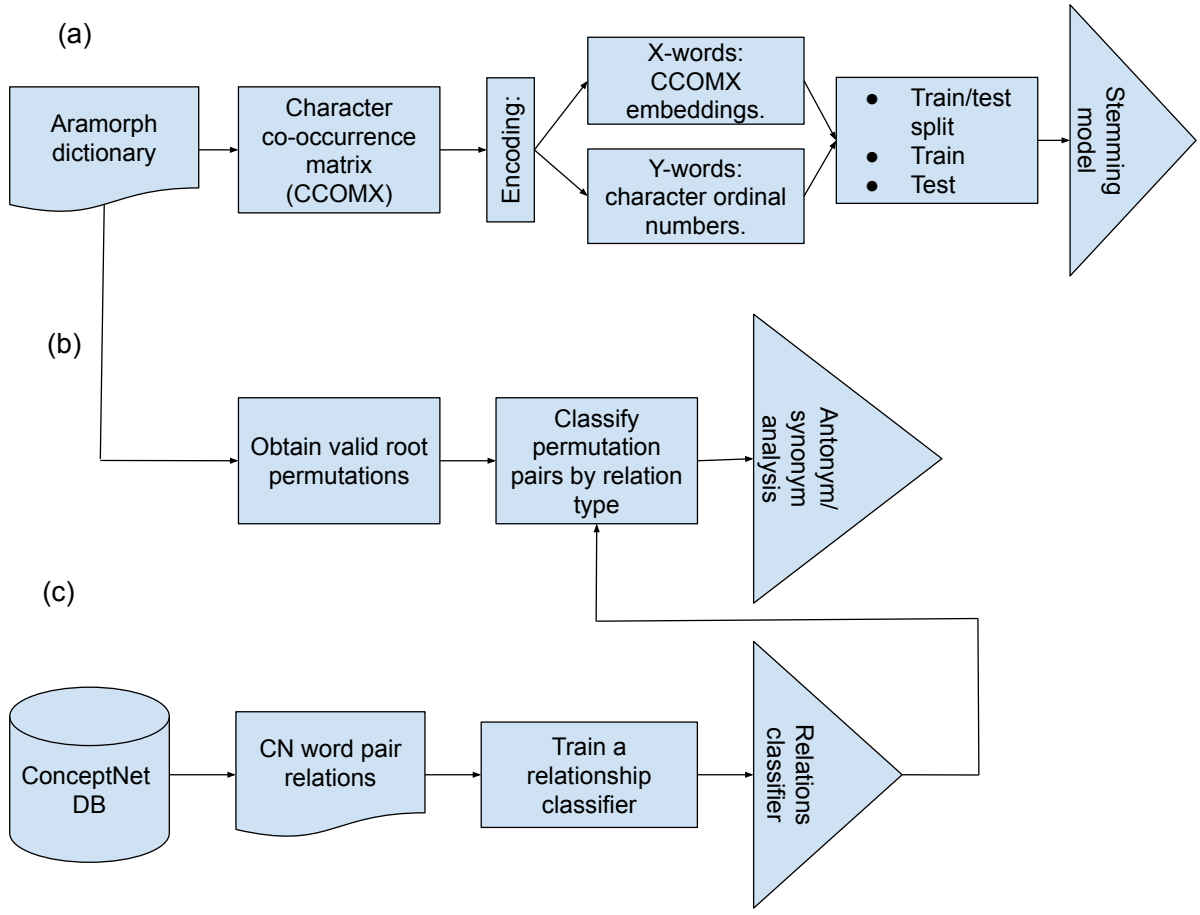


Fig. 1. The approach: (a) training the Arabic stemmer, (b) analyzing antonyms/synonyms amongst root character permutation pairs, and (c) training a word-pair relationship classifier

little attention in the Arabic linguistics literature is called *al-ʔi ʃtiqaaq al-kabiir* (الاشتقاق الكبير) 'the big derivation'; it concerns the idea that a root can have different permutations (recombination of

the root radicals) while still sharing a common meaning. Consider, for example, Table 2 (adapted from [16]: 153).

Essien [16], drawing on Ibn Jinnī [20], contends that the root k-l-m (ك ل م) and all of its permutations given in Table 2 convey the meaning of force or intensity.

Taking *al-ʔiftiqaaq al-kabiir* الاشتقاق الكبير 'the big derivation' as a point of departure, we have observed that some root permutations do not broadly result in a common meaning implied by or recoverable from the meanings/interpretations of the words, as shown in Table 2, but rather exhibit very specific semantic relations, particularly, synonymy and antonymy, to be discussed later.

Regarding language modeling, Larkey and Connell [23] participated in the TREC10 cross-language track, having no Arabic speakers among researchers and no experience with Arabic. The authors proposed using a character co-occurrence language modeling approach.

Having received moderate results, the authors improved dictionary construction, estimation, and smoothing in language models, ameliorated stemming and normalization, and expanded Arabic queries.

On the other hand, Khoja, Garside, and Knowles [21] described a morphosyntactic tag set from the perspective of ancient Arabic grammar that can precisely portray and embrace the entirety of the language. The tag set is based on the Semitic tradition of analyzing language.

The authors have demonstrated that in the tag set of the Arabic language, all the subcategories derive properties from the parent classes, thereby capturing the generalizations of this language.

Taghva et al. [37] implemented a stemmer for extracting roots for Arabic without a root dictionary, comparing it with Khoja's stemmer and other so-called "light" stemmers in document search tasks. This stemmer has proven to be as efficient as stemmers with a root dictionary.

Altabba, Al-Zaraee, and Shukairy [6] alternatively considered Qutuf - a system for marking parts of speech and morphological analysis of the Arabic language. Qutuf contains the morphological analysis component, which uses ending state automats and coinciding rules to analyze cliticalizations.

Farasa [1] is a precise and fast stemmer for the Arabic language based on SVM-rank using linear kernels. The stemmer showed efficiency in information retrieval (IR) and machine translation (MT) tasks. Farasa is on the same level or surpasses modern Arabic stemmers such as Stanford and MADAMIRA while demonstrating higher speed.

Last, but not least, Boudchiche et al. [9] developed AlKhalil Morpho Sys - a parser for morphosyntactic analysis of traditional Arabic words. This analyzer handles both Arabic words with voiced and unvoiced characters.

3 Data

3.1 Aramorph Arabic Morphological Analyzer

In this research, we have used data from Aramorph Arabic morphological analyzer [11] distributed under GNU General Public License (GPL) [39]. The data comprises a dictionary where for each of 82,160 Arabic word form entries, we have a vocabulary form, category, gloss (basically, an English translation), POS tag, lemma, and root. We dropped the entries with an empty root field to end up with 72,424 records.

The entries in the data table were transliterated to Latin characters, and we had to convert them back to original Arabic characters using the transliteration table [10] to be assessed by native Arabic speakers.

The Arabic word roots are known to possess abstract meanings that are modified by the patterns with which the roots combine.

The patterns sometimes consist only of vowels represented by diacritic characters [2] that are

[2] Instead, diacritic signs are used: *fatħa* (فتحة), *ḍamma* (ضمة), and *kasra* (كسرة).

Table 2. Root permutations and meanings

Root Permutations		Meanings
k-l-m (ك ل م)	kalm (كَلم) kalaam (كَلام)	Wound - the intense pain associated with it. Speech - the force produced when making an utterance.
k-m-l (ك م ل)	kaamil (كامِل)	Finished, complete – when something is completed, it is stronger and more powerful than something that is incomplete.
l-k-m (ل ك م)	lakm (لَكم)	Blow, punch - it is done with force.
m-k-l (م ك ل)	biʔr makuul (بِئر مَكول)	Dried up well - when a well dries up, it brings intense pain to the inhabitants.
m-l-k (م ل ك)	mulk (مُلك)	Possession – having power over others; authority and power.

Table 3. Unique values in the data table of the total of 72,424 records

Column	Unique values
FORM	39,586
VOC_FORM	48,765
CAT	156
GLOSS	32,001
POS	114
LEMMA	32,472
ROOT	4,094

Table 4. Number of the Arabic words in ConceptNet knowledge base by relation type label

Relation	Number of occurences
EtymologicallyRelatedTo	4,756
EtymologicallyDerivedFrom	656
Synonym	612
ExternalUrl	325
RelatedTo	298
DerivedFrom	280
Causes	250
Antonym	172
SimilarTo	19
CausesDesire	12
IsA	8
Desires	4
MotivatedByGoal	3
HasSubevent	2
HasFirstSubevent	2
DistinctFrom	2
NotDesires	1
SymbolOf	1
HasProperty	1
Total	7,404

generally invisible in writing; for instance, the word *fataha* 'he opened' is spelled as *fth* (فتح) with the *fatha* (= the vowel *a*) diacritics absent.

This, as a result, may often lead to the creation of homonyms (words having the same spelling but different meanings); for example, the words *fataha* 'he opened', *futiha* 'was opened', and *fath* 'opening' all are spelled as *fth* (فتح) in Arabic but their vowel diacritics and meanings can easily be recoverable from context.

In our dataset, we have primarily three-letter roots (3,170), and a small amount of four-letter roots (610) and two-letter roots (313); see Fig. 2.

3.2 ConceptNet Database

ConceptNet knowledge base contains 7,404 Arabic words with identified relations to other words out of 163,085 Arabic words in total, among which the most popular types are EtymologicallyRelatedTo, EtymologicallyDerivedFrom, and Synonym comprising more than 80% of all the words with a relation label present; see Table 4.

4 Methodology

4.1 Word Morphological Normalization

The means of word morphological normalization are usually Stemming and lemmatization, which are used to extract roots or bring the word to its initial indefinite form. These methods allow for a significant vocabulary reduction in corpora, as you can see in Table 3, just 4,094 roots correspond to the 48,765 vocabulary word forms.

Definition 4.1. In Arabic, a root consisting of two, three, or four consonants carries an abstract meaning as in *f-t-ħ, ḍ-r-b, n-w-m having to do with 'opening', hitting', and 'sleeping', respectively. The specific meaning of the root can be determined or modified based on the pattern the root combines with, as mentioned earlier [30, 29, 43, 19, 16, 42, 25, 12].*

Definition 4.2. A stem in Arabic is the third-person masculine singular past form of the verb, which is the base form of a word. This entails a reduction of derivational and inflectional morphemes, such as active and passive participle affixes or person, number, and gender affixes. Accordingly, the stem of *maktub-at* (مـكـتـوبـة) 'written' which is composed of the root *k-t-b*, the passive participle derivational morpheme *ma-u*, and the feminine singular inflectional morpheme -at should be *kataba* (كتب) [3].

While stemming is the reduction of the word to its base form, lemmatization in our analysis is the reduction of the word to its root, which is considered a label for large sets of related words [22]. For instance, the lemma of the words *kitaab* (كتاب)'book' , *maktuub-at* (مكتوبة) 'written-FM' , and *ya-ktub-uun* (يكتبون) 'they write' is the root *k-t-b* '(ك ت ب) [27, 24].

4.1.1 Stemming

Definition 4.3. Stemming, in our analysis, as mentioned earlier, is to reduce the lexical item to its verbal base form (the third person masculine singular past verb), removing any of its derivational and inflectional morphemes; see Table 5 for examples.

Table 5. Stemming example

Word	Meaning	Stem
darasa (دَرَسَ)	'he studied'	
darasat (دَرَسَت)	'she studied'	drs (درس)
madrasat (مَدرَسَة)	'school'	
diraasat (دِراسَة)	'studying'	

4.1.2 Lemmatization

Definition 4.4. Lemmatization, as mentioned above, is bringing the word to its root.

Highly Language Independent Word Lemmatizer (HLIWL) algorithm can be described as follows [2]:

1. Given the dictionary of Word form-lemma pairs, we assign them as independent (X) and dependent (Y) variables for applying the machine learning approach;

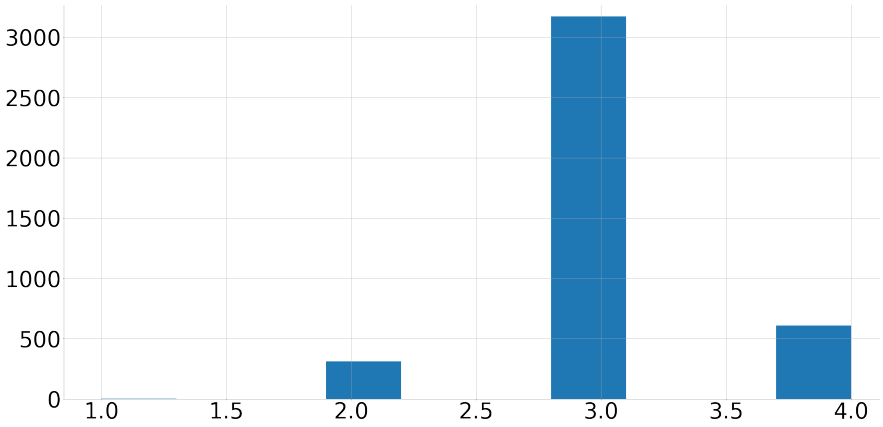


Fig. 2. Root lengths distribution

2. Preparing the character co-occurrence matrix where each row or column will serve as an embedding vector for the corresponding symbol;
3. Encoding the words in X by the character embeddings producing the vectors of the length of the longest word in the corpus and flattening it;
4. Words in Y we encode by the character ordinal number to carry on multiclass classification task;
5. Splitting the dataset $90/10\%$ for training and testing;
6. Training the Random Forest Classifier model employing the bootstrapping technique, 10 estimators, and using entropy as a criterion;
7. Testing the model.

4.2 Word Root Character Permutations

The method includes making all possible permutations of the characters in the roots and checking if the permutations are present in the dataset as roots and, thus, are valid. Further, we aimed to check for semantic links between the permutation variants.

In Islamic Shi'a and Sufi sources, such as the "Mystical commentary to the Qur'an, ascribed to Jafr Al-Sadiq" [4], there are four levels of understanding of sacred scripts:

— [ʕibɑːrah] (عبارة): literal expression for the commonality (ʔawaam).

— [ʔiˈʃɑrə] (إشارة): allusion for the elite (xawaaṣ).

— [laṭaaʔif] (لطائف): subtleties for the friends of Allah (SWT) (ʔawlijaaʔ).

— [ħaqaaʔiq] (حقائق): deepest realities for the prophets (ʔanbijaaʔ).

The number of all possible re-combinations of characters in a word of length n is given in (1).

$$N = n!, \tag{1}$$

where N is the number of all possible combinations, and n is the length of a word.

Example 4.1. For a three-letter root "abc" we have six $(3! = 6)$ non-repeating character permutations: 1) abc, 2) acb, 3) bac, 4) bca, 5) cab, and 6) cba (the reverse is a special case of character permutations).

4.3 Word Relations

In our work, we used the corpora provided by ConceptNet 5.5[3], which is an open multilingual graph of general knowledge base [36]. The knowledge base describes 34 types of relations between words including[4]:

— **RelatedTo**: when words have some positive relationship, but the nature of this relationship cannot be identified yet (ex. *learn* $->$ *erudition*).

— **FormOf**: when a word is an inflected form of another word (ex. *slept* $->$ *sleep*).

— **PartOf**: a *meronym* relationship in WordNet [17], when the word is a part of something more complex expressed by another word (ex. *wheel* $->$ *car*)

— **Synonym**: when words have similar meanings (ex. *good* <> *nice*).

— **Antonym**: when the words have opposite meanings (ex. *good* <> *bad*).

— **EtymologicallyRelatedTo**: when words have common origins (ex. *ecòle* fr. <> *okul* tr.).

— **EtymologicallyDerivedFrom**: when a word is derived from another word (ex. *circle* $->$ *church*, *circus*).

We decided to focus on two types of word relations, *Synonymy* and *Antonymy*, because they have a straightforward semantic link between words: the same or opposite meanings. All other types of relations are either too common, for example, etymological relations, or too rare, like *HasProperty*, *SimilarTo*, or *SymbolOf*.

4.3.1 Word Relation Machine Learning Classification Model

Building a classifier model requires preliminary steps such as Vectorization, Dataset balancing, and modeling.

Vectorization To apply any Machine Learning (ML) algorithm to train, test, and predict the class of two-word relations, we need to convert the words from a symbolic form to a vector form because ML algorithms work with numbers, not letters, characters, or words. We used the pre-trained FastText model for the Arabic language[5] [8, 18]. The choice of the vectorization method is justified by the fact that the FastText is the only model that focuses on a word as a collection of syllables, and this is precisely what we are looking for, as this works better with Arabic roots. However, we leave it for future work to experiment with other language models and vectorization techniques.

Dataset balancing As we saw in Table 4, the class distribution is highly unbalanced as more than 64% of all data points belong to one class, and all other classes comprise less than 9% each. Thus, to avoid classifier bias towards the majority class, we need to balance it using one of the following ways [13]:

— **Undersampling**: Randomly removing the items from majority classes to match the number of minority class items.

— **Oversampling**: Randomly reproducing (copy) minority class items to match the number of majority class items.

— **Synthetic Minority Oversampling**: Randomly generating minority class items similar to the original minority class data in Vector Space, using the K-Nearest Neighbor (KNN) algorithm.

We chose the Synthetic Minority Oversampling Technique (SMOTE) because undersampling reduces our initially small dataset of just 7,404 data points to merely 1,376, and oversampling reproduces existing data points, giving no additional information but instead decreasing the variance. At the same time, SMOTE artificially generates new valuable information for model training.

[3] https://conceptnet.io/

[4] https://github.com/commonsense/conceptnet5/wiki/Relations

[5] https://fasttext.cc/docs/en/crawl-vectors.html

Modeling For modeling, we selected `CatBoost` (gradient boosting on decision trees) algorithm because it shows superior performance even with the default parameters (improved accuracy and fast prediction) and works with word labels without the need for prior encoding of target categorical variables [28]. Although we used only one classification algorithm in the current work, we reserved the model selection procedure for future works.

4.3.2 Word Relation Statistical Classification Model

Along with the ML approach for word relation classification, we implemented the statistical approach, which was to define the thresholds for the distance between a pair of words in the vector space to distinguish between the "*Synonym*" and "*Antonym*."

After we measured the cosine distances in Vector Space (see section 4.3.3) between word pairs labeled for the type of semantic relation, we grouped the distances by the relation class and calculated the averages and standard deviations of the distances; see Table 6.

As we can see from Fig. 3, statistically, we can more or less distinguish only the Synonym type of the semantic relationship among words because it has fewer overlaps of the range within one standard deviation from the mean, where around 67% of observations must fall under the normal distribution.

4.3.3 Vector Space Model

The method is based on a vector representation of objects, in general, to apply general Linear Algebra operations and compute the proximity (cosine or Euclidean distance measures) between the objects and get corresponding intuition on their interrelation and its strength [31].

Definition 4.5. Cosine distance is a measure of proximity of two vectors characterized by the cosine of the angle between them and closely related to the term of cosine similarity; see (2) and (3) [34].

$$\text{cos\_sim} = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

where A_i and B_i are components of vector A and B, respectively.

$$\text{cosine distance} = 1 - \text{cos\_sim}. \quad (3)$$

Definition 4.6. Euclidean distance, as the name implies, is the distance between two points in Euclidean space, which can be calculated using the Pythagorean theorem and Cartesian coordinates of the points; see (4) [35]:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2} \quad (4)$$

where $p_1 \ldots p_n$ and $q_1 \ldots q_n$ are the dimensions of points p and q respectively.

Definition 4.7. Vector representation of objects o_i.

$$\begin{aligned} o_i &= (w_{1,i}, w_{2,i}, \ldots, w_{t,i}) \\ &\ldots \\ o_n &= (w_{1,n}, w_{2,n}, \ldots, w_{t,n}) \end{aligned} \quad (5)$$

Each vector dimension corresponds to an object's feature; in our case, these are the features extracted by the FastText language model for Arabic.

4.4 Evaluation

4.4.1 Accuracy

The accuracy metric is the portion of predictions indicating that a classification model is correct. In classification, accuracy has the following formula [26]:

$$\text{Accuracy} = \frac{\text{true positives + true negatives}}{\text{total number of examples}}, \quad (6)$$

where $true\ positives$ is the number of items correctly identified as the target class by a model, $true\ negatives$ is the number of items correctly classified not as target class, and $total\ number\ of\ examples$ is the sum of $true\,positives$, $true\,negatives$, $false\,positives$, and $false\ negatives$. Each class's accuracy metric is separately calculated using (6) for multiclass classification.

Table 6. Cosine distances averages, standard deviation (std), lower (low), and upper (high) boundaries within one standard deviation from the average and the range

Semantic relation	average	std	low	high	range
Antonym	0.78	0.18	0.60	0.96	0.37
Causes	0.78	0.17	0.61	0.96	0.34
CausesDesire	0.72	0.21	0.51	0.93	0.42
DerivedFrom	0.85	0.12	0.72	0.97	0.25
Desires	0.86	0.05	0.81	0.91	0.10
DistinctFrom	0.80	0.02	0.77	0.82	0.05
EtymologicallyDerivedFrom	0.81	0.15	0.66	0.97	0.30
EtymologicallyRelatedTo	0.67	0.25	0.42	0.93	0.50
ExternalUrl	0.67	0.28	0.39	0.95	0.56
HasFirstSubevent	0.87	0.07	0.80	0.93	0.13
HasSubevent	0.86	0.07	0.79	0.93	0.14
IsA	0.82	0.13	0.69	0.96	0.26
MotivatedByGoal	0.89	0.08	0.80	0.97	0.16
RelatedTo	0.76	0.22	0.54	0.98	0.43
SimilarTo	0.78	0.15	0.63	0.93	0.30
Synonym	0.24	0.38	-0.14	0.61	0.75

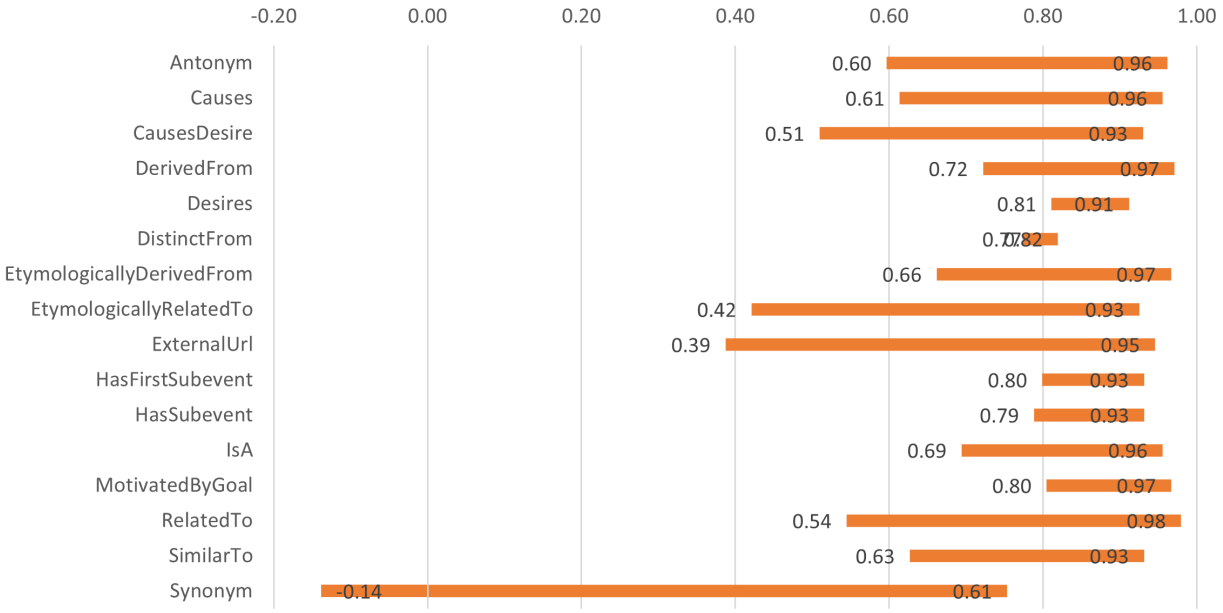


Fig. 3. Cosine similarity value ranges by class of the word semantic relation

4.4.2 F1 Score

For imbalanced datasets, the accuracy metric is not a proper measure as it can be maximized simply by classifying all the items for the majority class. Therefore, other quality metrics should be used to account for the issue, such as the F1 score. It can be defined as the harmonic mean of

Precision and Recall [15, 32]:

$$\text{F1 score} = \frac{2PR}{P+R}, \tag{7}$$

where P is the Precision and R is the Recall, see (8) and (9), respectively.

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives +false positives}}, \tag{8}$$

where $false\ positives$ is the number of Type I errors when a target class is not classified as positive.

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives +false negatives}}, \tag{9}$$

where $false\ negatives$ is the number of Type II errors when a target class is misclassified as negative.

5 Experiments

5.1 Stemming and Lemmatization

In this research, we employed our approach of language-independent word normalization using character embeddings and multi-class multi-label `RandomForest` classifier [2]. The technique was applied to stemming and lemmatization, using transliteration and original Arabic script. We called the model HLIWL abbreviating the original article title initial words for "Highly Language Independent Word Lemmatizer (HLIWL)".

HLIWL algorithm can be described as follows[6] [2]:

1. Given the dictionary of Word form-lemma pairs, we assigned them as independent (X) and dependent (Y) variables for applying the machine learning approach;
2. Data cleansing steps:
   — Cleaning or imputing the empty record fields.
   — Eliminating trailing spaces, special symbols, and punctuation.
3. Data preparation steps:
   — Preparing the character co-occurrence matrix where each row or column serves as an embedding vector for the corresponding symbol;
   — Encoding the words in X by the character embeddings producing the vectors of the length of the longest word in the corpus and flattening it;
   — Encoding words in Y by the character ordinal number to carry on a multiclass classification task;
4. Training and testing steps:
   — Splitting the dataset $90/10\%$ for training and testing;
   — Training the `RandomForest` Classifier model employing the bootstrapping technique, 10 estimators, and using entropy as a criterion;
   — Testing the model.

Definition 5.1. Word form is any inflected lexeme derived from its indefinite form for verbs or singular and masculine form for nouns.

Example 5.1. "*running*" and "*ran*" are the word forms of the verb "*to run*" or lemma "*run*".
"*horses*" is a plural word form of the noun "*horse*".

We divided the dataset described in section 3 into the train/test sets, leaving for the test a ratio of 0.1 or 7,243 randomly selected records to train our models on the train set and test it together with other stemmers on the test set.

After running the experiments using our approach, we compared our results with the following Arabic Stemmers[7] on the test set:

1. Assem Chelli[8] [14].
2. Al Khalil Morpho Sys Stemmer [9].

[6] https://github.com/iskander-akhmetov/Highly-Language-Independent-Word-Lemmatization-Using-a-Machine-Learning-Classifier

[7] https://github.com/MaafiHanene/Arabic-Stemmers

[8] `https://github.com/assem-ch/arabicstemmer`

3. Arabic-Stemming-Toolkit (AST) [5].2
4. ArabicProcessingCog[9].
5. FARASA[10] [1].
6. Lucene Arabic Analyzer[11].
7. ISRIStemmer[12] [37, 7].
8. Qutuf[13] [6].
9. Shereene Khoja Stemmer [23, 21].
10. Tashaphyne: Arabic Light Stemmer[14] [44].

5.2 Classifying Word Relations

Two different approaches were used to classify pairs of words based on their semantic relation: the ML approach and the statistical one.

5.2.1 ML Approach

The experiment was set up in the following way:

1. We took the data extracted from the ConceptNet database; see section 4.3, with pairs of words (the starting word and ending word) labeled for the semantic relations.
2. We vectorized the Arabic words with the FastText pre-trained Arabic language model.
3. We concatenated the vectors obtained for both words in a pair to use it as an independent variable (X).
4. We merged word relationship classes with small amount of incidences (see Table 4, "*HasSubevent*", "*Desires*", "*CausesDesire*", "*NotDesires*", "*MotivatedByGoal*", "*HasFirstSubevent*", "*SymbolOf*", "*IsA*", "*DistinctFrom*", "*SimilarTo*", "*HasProperty*", "*ExternalUrl*") to a new "*Minor*" class; see Table 8.
5. We set word relationship class labels as the target variable (Y).
6. We split the dataset to train/test subsets leaving 33% for the test.
7. We trained a `CatBoost` classifier model on the train set.
8. We tested the trained model on the test set.

5.2.2 Statisical Approach

For the statistical approach, we reused steps 1-2 from the ML approach and continued as follows:

1. We calculated the cosine distances between all of the 7,404-word pairs.
2. We grouped and took the average of the distances by the relationship class.
3. We devised a rule to classify semantic relationship's antonym or synonym types.
4. We tested the rule on the whole dataset.

6 Results

6.1 HLIWL Model Performance

The HLIWL model showed the best results for the stemming task, resulting in 0.7469/0.9514 on test/train accuracy scores. The model yielded poor results for the Lemmatization task 0.2713 and 0.2706 test set performance on Arabic and transliterated characters, respectively. The approach also appears to work worse on the transliterated characters than on the original Arabic ones for both stemming and lemmatization test set performance; see Table 7.

Nevertheless, the HLIWL model appears to be outperforming all of the stemmer models listed in section 5.1, scoring 0.1959 points for accuracy, which is higher than the second place winner, the Lucene Arabic Analyzer model on the test set; see Table 9 and 10.

[9] https://github.com/disooqi/ArabicProcessingCog
[10] `https://alt.qcri.org/farasa/`
[11] https://github.com/msarhan/lucene-arabic-analyzer
[12] `https://www.nltk.org/_modules/nltk/stem/isri.html`
[13] `https://github.com/Qutuf/Qutuf`
[14] https://pypi.org/project/Tashaphyne/

Table 7. HLIWL Arabic model performance for stemming and lemmatization

	Test accuracy	Train accuracy	Max word len
Stemmer			
HLIWL (arabic script)	0.7469	0.9514	12
HLIWL (translit)	0.7280	0.9510	
Lemmatizer			
HLIWL (arabic script)	0.2713	0.6347	18
HLIWL (translit)	0.2706	0.6349	

Table 8. ConceptNet Data Base word pair relationship class labels distribution after merging underrepresented classes to a "*Minor*" class

Relation	Number of occurences
EtymologicallyRelatedTo	4,756
EtymologicallyDerivedFrom	656
Synonym	612
Minor	380
RelatedTo	298
DerivedFrom	280
Causes	250
Antonym	172
Total	7,404

Table 9. Comparing stemming performance on 7,243 words from the test set (numbers in bold indicate leaders)

Stemmer	Test accuracy
This work	
HLIWL (Arabic script)	**0.7469**
HLIWL (translit)	0.7280
Other stemmers	
Lucene Arabic Analyzer	**0.5510**
Shereene Khoja Stemmer	0.5494
ISRIStemmer	0.5397
ArabicProcessingCog	0.2361
Arabic-Stemming-Toolkit AST	0.2358
Tashaphyne: Arabic Light Stemmer	0.2332
Assem's Arabic Light Stemmer	0.2310
FARASA	0.2303
Al Khalil Morpho Sys Stemmer	0.2195
Qutuf	0.1923

6.2 Word Pair Relationship Classifier Results

Here, we described the results of experiments on building classifier models for the word pair semantic relationship using ML and statistical inference. As we can see, both models showed poor performance, which can be attributed to two facts:

— Word semantic relationship classes are difficult to predict using distributional semantics language model embeddings [38]. For example, antonyms are hardly distinguishable from other types of semantic relationships such as "*Causes*", "*SimilarTo*" or "*RelatedTo*"; see Fig. 3 and Table 6.

— The dataset we used contains single words and their features outside of the context to derive additional features as Part of Speech (POS) Tag, which might be helpful [33].

Nevertheless, the low-performing classifier models are compensated by the expert opinion in linguistics and the Arabic native speaker, who manually checked the classification.

6.2.1 ML Approach

The ML approach used the `CatBoost` algorithm to build a classifier model. The model was trained on the dataset oversampled using the SMOTE method and tested on the data with the original class composition structure with a major class (63%) of "*EtymologicallyRelatedTo*"; see Table 11 and 12 for the confusion matrix and classification report.

As shown in Table 12, the class of "*antonyms*" is highly misclassified, yielding merely 0.31 accuracy and 0.43 F1 scores in its class, while the class

Table 10. Examples of stemming results by the compared models

Word form	إختلس	ترقوص	عرك	مخالب	مجمر	لجوئ	إستتمام	ستترد	منغاظ	بطان
Root	خلس	رقص	عرك	خلب	جمر	لجأ	تمم	رد	غيظ	بطن
Stemmer results										
Assem's Arabic Stemmer (snowball)	اختلس	ترقوص	عرك	مخالب	مجمر	لجوء	استتمام	ستترد	منغاظ	طان
Al Khalil Morpho Sys Stemmer	إختلس	ترقوص	عرك	مخالب	مجمر	لجوئ	إستتمام	ستترد	منغاظ	بطان
Arabic Stemmer Toolkit	اختلس	ترقوص	عرك	مخالب	مجمر	لجوئ	استتمام	ستترد	منغاظ	بط
ArabicProcessingCog	إختلس	ترقوص	عرك	مخالب	مجمر	لجوئ	إستتمام	ستترد	منغاظ	بط
FARASA	إختلس	ترقوص	عرك	مخلب	مجمر	لجوئ	إستتمام	رد	منغاظ	بطان
Lucene Arabic Analyzer	خلس	رقوص	عرك	خلب	جمر	لجأ	تمم	ردد	غيظ	بطن
ISRIStemmer	خلس	رقص	عرك	خلب	جمر	لجئ	تمم	ترد	نغظ	بطن
Qutuf	إختلس	ترقوص	عرك	مخالب	مجمر	لجوئ	إستتمام	ترد	منغاظ	بطان، طان
Shereene Khoja Stemmer	خلس	رقوص	عرك	خلب	جمر	لجأ	تمم	ردد	غيظ	بطن
Tashaphyne	إختلس	رقوص	عر	خالب	جمر	جوئ	إستتمام	رد	غاظ	ط
HLIWL (this work)	خل	رقوص	عرك	خلط	جمر	لجأ	تمم	رود	نغم	بطن

Table 11. ML approach confusion matrix

Actual\Predicted	Minor	Syn.	Etym. Derived From	Ant.	Derived From	Etym. Related To	Causes	Related To	Support
Minor	**20**	2	18	1	1	83	0	0	125
Syn.	1	**123**	16	1	5	70	1	0	217
Etym.Derived From	0	1	**117**	2	10	92	2	2	226
Ant.	1	3	20	**19**	3	15	1	0	62
Derived From	1	0	24	1	**48**	17	0	1	92
Etym. Related To	8	39	62	2	9	**1411**	2	8	1541
Causes	2	0	10	0	9	30	**39**	0	90
Related To	0	1	8	1	3	64	1	**13**	91
Total	33	169	275	27	88	1782	46	24	2444

of "*synonyms*" is better, but still slightly more than half of it is classified correctly (0.57 accuracy and 0.64 F1 scores). The class weighted average accuracy, and F1 scores are 0.73 and 0.71, respectively, primarily attributed to the dominant "*EtymologicallyRelatedTo*" class.

6.2.2 Statistical Approach

After looking at the ranges in Fig. 3 and distributions of distances between the pairs of "synonyms" and "antonyms" in Vector Space; see Fig. 4, we empirically inferred following rules:

— **Synonyms**: if the distance is less than or equal to 0.1, classify a pair of words as "*synonyms*".

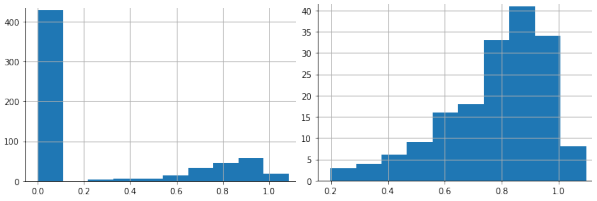
— **antonyms**: if the distance is less than or equal to 1.2 and greater than or equal to 0.9, then classify a pair of words as "*Antonyms*."

— **Unknown**: All other distance ranges are classified as "Unknown," which includes the remaining semantic relationship classes.

The classification report in Table 13 shows that the statistical inference approach performs better on the class of "synonyms" than the ML approach,

Table 12. ML approach classification report

	Accuracy	Precision	Recall	F1-score	Weight
Minor	0.16	0.61	0.16	0.25	5.1%
Synonym	0.57	0.73	0.57	0.64	8.9%
EtymologicallyDerivedFrom	0.52	0.43	0.52	0.47	9.2%
Antonym	0.31	0.70	0.31	0.43	2.5%
DerivedFrom	0.52	0.55	0.52	0.53	3.8%
EtymologicallyRelatedTo	0.92	0.79	0.92	0.85	63.1%
Causes	0.43	0.85	0.43	0.57	3.7%
RelatedTo	0.14	0.54	0.14	0.23	3.7%
Weighted average	0.73	0.72	0.73	0.71	100.0%



(a) Synonym pairs distances **(b)** Antonym pairs distances

Fig. 4. Word pair distance distributions

and the weighted averages for accuracy are also higher.

Table 13. Statistical approach classification report

	precision	recall	f1-score	support
Antonym	0.03	0.29	0.06	172
Synonym	0.86	0.70	0.77	612
Unknown	0.95	0.77	0.85	6,620
accuracy			0.75	7,404
macro avg	0.61	0.59	0.56	7,404
weighted avg	0.92	0.75	0.83	7,404

6.3 Root Characters Permutation Variants Semantic Links

In section 2, we discussed the semantic properties of the permutations of roots described in previous research. Notably, we showed that the concept of *al-ʔiʃtiqaːq al-kabiːr* (الاشتقاق الكبير) 'the big derivation' concerns that idea that root permutations result in a common meaning implied by the meanings/interpretations of the words, as shown in Table 1 Essien [16].

This is, however, not the whole story. Upon closer scrutiny of root permutations, we have observed that root permutations can specifically exhibit two very specific semantic relations: synonymy and antonymy. For example, the permutations of the roots in Table 14 produce synonyms.

On the other hand, the permutation of the roots in Table 15 gives rise to antonyms.

7 Discussion

What is the difference between EtymologicallyRelatedTo and EtymologicallyDerivedFrom, and how do we distinguish them? When a word is derived from another word, it is already related to that word, isn't it? So EtymologicallyRelatedTo must be a Hyperonym or Synonym of EtymologicallyDerivedFrom.

Both of our models for classifying word relationships showed poor performance. Thus, there is room for development in this area of research, which we hope to explore soon.

What other types of word semantic relations are worth exploring? Some interesting relation candidates to find between permuted roots are:

Table 14. Synonym examples of character permutations in roots

Semantic relation	Root	Root permutation	Meaning
Synonymy	m-w-h (م و ه)	w-h-m (و ه م)	deceiving
	k-m-t (ك م ت)	k-t-m (ك ت م)	suppressing
	r-ḍ-ħ(ر ض ع)	ḍ-r-ħ(ض ر ع)	suckling
	ħ-b-r (ع ب ر)	ħ-r-b (ع ر ب)	expressing
	d-ʃ-r (د ش ر)	ʃ-r-d (ش ر د)	straying
	ʃ-ṭ-ħ(ش ط ح)	ʃ-ħ-ṭ (ش ح ط)	exaggerating
	ṣ-f-ħ(ص ف ع)	ħ-f-ṣ (ع ف ص)	hitting
	m-d-ħ(م د ح)	ħ-m-d (ح م د)	praising
	q-ḍ-b (ق ض ب)	q-b-ḍ (ق ب ض)	catching
	k-ħ-l (ك ح ل)	ħ-l-k (ح ل ك)	blackening
	x-r-b-ṭ (خ ر ب ط)	b-x-r-ṭ (ب خ ر ط)	mixing up

Table 15. Antonym examples of character permutation in roots

Semantic relation	Root	Meaning	Root permutation	Meaning
Antonymy	m-h-d (م ه د)	fixing	h-d-m (ه د م)	destroying
	ʃ-r-ħ(ش ر ح)	widening	ħ-ʃ-r (ح ش ر)	narrowing
	f-l-q (ف ل ق)	opening	q-f-l (ق ف ل)	closing
	k-l-ħ(ك ل ح)	bleaching	ħ-l-k (ح ل ك)	blackening
	r-b-ṭ (ر ب ط)	tying	b-r-ṭ (ب ر ط)	untying
	ħ-s-n (ح س ن)	improving	n-ħ-s (ن ح س)	worsening
	w-d-ʕ(و د ع)	leaving	ʕ-w-d (ع و د)	returning
	r-ħ-m (ر ح م)	having mercy	ħ-r-m (ح ر م)	oppressing
	b-r-k (ب ر ك)	blessing	k-r-b (ك ر ب)	distressing
	ʃ-k-r (ش ك ر)	thanking	ʃ-r-k (ش ر ك)	setting a trap
	ʤ-l-ʤ-l (ج ل ج ل)	sound-booming	l-ʤ-l-ʤ(ل ج ل ج)	stammering

— **IsA**: when A is a *hyponym* of B, or when A is the special case of B. For example, *man* $->$ *human*, *bulldog* $->$ *dog*, *limo* $->$ *car*.

— **PartOF**: When A is a *meronym* or a constituent of B. For example, *roof* $->$ *house*, *engine* $->$ *jet*, *sail* $->$ *boat*. The opposite relationship type is a **HasA**.

— **UsedFor**: If A is used as a purpose for B. For example, *plane* $->$ *fly*, *ship* $->$ *sail*, newspaper $->$ read the news.

— **Causes**: When A is the reason for B. For example, *hurt* $->$ *pain*, *study* $->$ *degree*, *exercise* $->$ *strength*.

— **CapableOf**: if A can do B. For example, *horse* $->$ *kick*, *wolf* $->$ *bite*, *cheetah* $->$ *run*.

8 Conclusion

We showed in this paper that the HLIWL approach performs fairly well on the Arabic word-stemming

task, reaching an accuracy of 0.75 and surpassing other Arabic stemmers with a good margin; see Table 9. This means there is still room for development in the research on Arabic stemming tasks.

The word relations classification task is yet to be researched more closely as we have achieved decent results in this work; see Table 12 and 13. Poor classification accuracy of synonyms, especially antonyms, may be attributed to our dataset's few examples of such relations and the dominance of the *EtymologycallyRelatedTo* class. Crucially, we showed that permutating some Arabic roots produces a meaning somehow related to the meaning of the roots before the premutation, such as synonymy and antonymy.

For future research projects, we plan to:

1. Continue the research on the Arabic stemmer development.
2. Repeat the root character permutations on larger corpora.
3. Experiment with language models and vectorization techniques to build word relation classifiers.
4. Implement the model selection procedure to choose the best-performing word relations classifier using cross-validation.

Acknowledgments

This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP23486904).

List of abbreviations

— AST: Arabic Stemming Toolkit.

— F: Feminine.

— HLIWL: Highly Language Independent Word Lemmatizer.

— KNN: K-Nearest Neighbors.

— M: Masculine.

— ML: Machine Learning.

— NLP: Natural Language Processing.

— NOM: Nominative.

— POS: Part of Speech.

— SMOTE: Synthetic Minority Oversampling Technique.

— STD: Standard Deviation.

References

1. **Abdelali, A., Darwish, K., Durrani, N., Mubarak, H. (2016).** Farasa: A fast and furious segmenter for Arabic. 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 11–16.
2. **Akhmetov, I., Pak, A., Ualiyeva, I., Gelbukh, A. (2020).** Highly language-independent word lemmatization using a machine-learning classifier. Comput. sist., Vol. 24, No. 3.
3. **Al-Abweeny, W. W., Zaid, N. A. (2018).** Arabic stemmer system based on rules of roots. International Journal of Information Technology and Language Studies, Vol. 2, pp. 19–26.
4. **Al-Sadiq, J., Mayer, F., Nwyia, P. (2011).** Spiritual Gems: The Mystical Qur'an Commentary Ascribed to Jafar al-Sadiq as contained in Sulamls Haqaiq al-Tafsir from the text of Paul Nwyia. The Fons Vitae Qur'anic Commentary Series. Fons Vitae.
5. **Almusaddar, M. Y. (2014).** Improving Arabic light stemming in information retrieval systems. Islamic University,Faculty of Engineering Computer, Gaza, Palestine Research, Vol. 1.
6. **Altabbaa, M., Al-Zaraee, A., Shukairy, M. (2010).** An Arabic Morphological Analyzer (Including Stemming and Root Extraction) and Part-Of-Speech Tagger as an Expert System. PhD dissertation, Arab International University, Damascus, Syria.

7. **Bird, S., Klein, E., Loper, E. (2009).** Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".
8. **Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., Douze, M., Jegou, H. (2016).** Fasttext.zip: Compressing text classification models.
9. **Boudchiche, M., Mazroui, A., Bebah, M., Lakhouaja, A., Boudlal, A. (2017).** Alkhalil morpho sys 2: A robust Arabic morpho-syntactic analyzer. Journal of King Saud University - Computer and Information Sciences, Vol. 29, pp. 141–146. DOI: 10.1016/j.jksuci.2016.05.002.
10. **Buckwalter, T.**, **.** Buckwalter Arabic Transliteration.
11. **Buckwalter, T. (2004).** Buckwalter arabic morphological analyzer version 2.0.
12. **Caspari, C. (1896).** A grammar of the Arabic language. Cambridge University Press.
13. **Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002).** SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, Vol. 16, pp. 321–357. DOI: 10.1613/jair.953.
14. **Chelli, A. (2018).** Assem's Arabic Stemmer. DOI: 10.6084/m9.figshare.7295690.v1.
15. **Chinchor, N. (1992).** Muc-4 evaluation metrics. Proceedings of the 4th Conference on Message Understanding, Association for Computational Linguistics, USA, pp. 22–29. DOI: 10.3115/1072064.1072067.
16. **Essien, H. (2014).** Structural study of Arabic morphology. PhD dissertation, Indiana University, Bloomington, USA.
17. **Fellbaum, C. (1998).** WordNet: An Electronic Lexical Database. Bradford Books.
18. **Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T. (2018).** Learning word vectors for 157 languages. DOI: 10.48550/ARXIV.1802.06893.
19. **Holes, C. (2004).** Modern Arabic: Structures, Functions, and Varieties. G - Reference,Information and Interdisciplinary Subjects Series. Georgetown University Press.
20. **Ibn Jinni, A. a.-F. U. (1952).** al-Khasa'is. Bayrut: Dar al-Huda.
21. **Khoja, S., Garside, R., Knowles, G. (2001).** An Arabic tagset for the morphosyntactic tagging of Arabic. PhD dissertation, Lancaster University, Lancaster, United Kingdom.
22. **Knowles, G., Don, Z. M. (2004).** The notion of a "lemma": Headwords, roots and lexical sets. International Journal of Corpus Linguistics, Vol. 9, No. 1, pp. 69–81. DOI: https://doi.org/10.1075/ijcl.9.1.04kno.
23. **Larkey, L. S., Connel, M. E. (2001).** Automatic information retrieval at umass in trec-10. The Tenth Text REtrieval Conference.
24. **Lestari, R., Fitriani, R., Nurlaili, D., Ismaiyah, L. N., Umam, K. F., Bimantara, B. (2019).** Root and stem in English and Arabic language. josar, Vol. 2, No. 1, pp. 1–14.
25. **McCarthy, J. J. (2018).** Formal problems in Semitic phonology and morphology. Routledge.
26. **Metz, C. E. (1978).** Basic principles of roc analysis. Seminars in Nuclear Medicine, Vol. 8, No. 4, pp. 283–298. DOI: https://doi.org/10.1016/S0001-2998(78)80014-2.
27. **Namly, D., Bouzoubaa, K., El Jihad, A., Aouragh, S. L. (2020).** Improving Arabic lemmatization through a lemmas database and a machine-learning technique. In **Abd Elaziz, M., Al-qaness, M. A. A., Ewees, A. A., Dahou, A.**, editors, Recent Advances in NLP: The Case of Arabic Language. Springer International Publishing, Cham, pp. 81–100. DOI: 10.1007/978-3-030-34614-0_5.
28. **Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A. (2017).** Catboost: unbiased boosting with categorical features. DOI: 10.48550/ARXIV.1706.09516.
29. **Ryding, K. (2014).** Arabic: A Linguistic Introduction. Cambridge University Press.

30. **Ryding, K. C. (2005).** A Reference Grammar of Modern Standard Arabic. Reference Grammars. Cambridge University Press. DOI: 10.1017/CBO9780511486975.

31. **Salton, G. (1962).** Some experiments in the generation of word and document associations. Proceedings of the December 4-6, 1962, Fall Joint Computer Conference, Association for Computing Machinery, New York, NY, USA, pp. 234–250. DOI: 10.1145/1461518.1461544.

32. **Sasaki, Y. (2007).** The truth of the F-measure. Teach Tutor mater.

33. **Scheible, S., Schulte im Walde, S., Springorum, S. (2013).** Uncovering distributional differences between synonyms and antonyms in a word space model. Proceedings of the Sixth International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Nagoya, Japan, pp. 489–497.

34. **Singhal, A. (2001).** Modern information retrieval: A brief overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24.

35. **Smith, K. (2013).** Precalculus: A Functional Approach to Graphing and Problem Solving. The Jones & Bartlett learning series in mathematics. Jones & Bartlett Learning.

36. **Speer, R., Chin, J., Havasi, C. (2016).** Conceptnet 5.5: An open multilingual graph of general knowledge. DOI: 10.48550/ARXIV.1612.03975.

37. **Taghva, K., Elkhoury, R., Coombs, J. (2005).** Arabic stemming without a root dictionary. International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, Vol. 1, pp. 152–157 Vol. 1. DOI: 10.1109/ITCC.2005.90.

38. **Thalenberg, B. (2016).** Distinguishing antonyms from synonyms in vector space models of semantics.

39. **The Aratools team (2022).** Aratools Arabic-English dictionary.

40. **Tolegen, G., Toleu, A., Mussabayev, R. (2022).** A finite state transducer based morphological analyzer for the kazakh language. 2022 7th International Conference on Computer Science and Engineering (UBMK), pp. 01–06. DOI: 10.1109/UBMK55850.2022.9919445.

41. **Toleu, A., Tolegen, G., Mussabayev, R. (2022).** Language-independent approach for morphological disambiguation. Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, pp. 5288–5297.

42. **Watson, J. (2002).** The Phonology and Morphology of Arabic. Oxford linguistics. Oxford University Press.

43. **Zawaydeh, B. (2003).** Janet c. e. watson (2002). the phonology and morphology of Arabic. oxford: Oxford university press. pp. v 307.. Phonology, Vol. 20, No. 2, pp. 280–283. DOI: 10.1017/S0952675703004548.

44. **Zerrouki, T. (2012).** Tashaphyne, Arabic light stemmer.

Article received on 22/08/2025; accepted on 18/11/2025.
**Corresponding author is Alexander Gelbukh.*