ARABIC NAMED ENTITY RECOGNITION BASED ON TREE-BASED PIPELINE OPTIMIZATION TOOL

[1]BRAHIM AIT BEN ALI, [1]SOUKAINA MIHI, [2]ISMAIL EL BAZI, [1]NABIL LAACHFOUBI

[1]Hassan First University of Settat, Faculty of Sciences and Techniques, IR2M Laboratory, Settat, Morocco

[2] Sultan Moulay Slimane University, National School of Business and Management, Beni Mellal, Morocco

E-mail: [1]aitbenali.brahim@gmail.com, [1]mihi.soukaina@gmail.com, [2]ismailelbazi@gmail.com, [1]n.laachfoubi@hotmail.fr

ABSTRACT

Named Entity Recognition (NER) is a clue task to improve automatic text processing, which is needed in a diverse variety of applications. NER techniques vary from hand-craft rules to machine learning approaches. As a human opting for a supervised learning algorithm, it is often difficult to choose the optimal machine learning (ML) model for a classification problem. Automated ML (AutoML) aims at automatically selecting, composing, and parameterizing machine learning algorithms in order to obtain optimal performance for a given task (dataset). In this paper, we apply an approach based on Tree-based Pipeline Optimization Tool (TPOT). This method uses genetic programming based on the tree structure to find the model and its hyperparameters that more closely predicts the class of Arabic named entities in the text comes from social media. The structure and parameters are fine-tuned to achieve the optimum performance of the machine's learning pipeline. Our method outperforms strong baselines. It reaches a novel state-of-the-art in the NER task.

Keywords: *Arabic Named Entity Recognition, NLP, Social Media, Machine Learning, Automl, Genetic Programming, Tree-Based Pipeline Optimization Tool* **(***TPOT).*

1. INTRODUCTION

Named Entity Recognition (NER) aims at recognizing mentions of rigid tags from texts that belong to predefined semantic types as person, location, organization, etc. [15]. NER serves not only as a stand-alone Information Extraction (IE) tool, but also plays an important role in a wide variety of Natural Language Processing (NLP) applications including information retrieval [22], [41], question answering [33], automatic text summarization [3], [43], machine translation [7], and knowledge base construction [19], etc.

It is stated that most of the research projects around this topic in the different languages have achieved a very high level of performance comparable to that of human subjects, especially in English [20].

Nowadays, where social media are omnipresent, dealing with informal forms is becoming increasingly crucial. In recent years, NER in social media has received considerable attention because of its many new challenges. Hence, a considerable number of studies have applied to NER in dialect Arabic and have not successively advanced state-of-the-art performance. This trend motivates us to develop a new approach that allows us to extract the named entities in the informal text from social media and to obtain a very considerable level of performance when comparing with existing systems.

The identification of NEs in the raw information is not an easy task in the Arabic language because the Arabic language is not capitalized. Resources such as gazetteers, dictionaries, POS markers, morphological analyzers are not freely accessible. There are many variations in spelling style. Work on NERs in Arabic languages, particularly colloquial Arabic, is a complex and challenging task as well as limited because of the lack of resources, but it started to emerge. That is why we developed a new system that allowed us to manipulate this kind of text.

Over the past few years, machine learning approaches (AutoML) have made significant progress in automatically finding machine learning solutions from a wide range of possible models. In general, a solution is usually created by selecting a model (e.g., random forest, SVM, etc.) and then set up these models by assigning hyperparameter values to them.

Currently, the recognition of the named entity is essentially based on the machine learning method. Most machine learning methods are applied in a specific framework, and they have to choose the algorithm and adjust the hyperparameters automatically. When the data set changes, the accuracy of the predictions is affected. It is particularly important to find a machine learning method that can automatically select features, algorithms, models, and pipelines. In practice, it is possible to predict the model well, but most people who use machine learning methods know that cross-validation is sometimes misleading and that the model is unknowable. These phenomena will reduce our confidence level in the model;

Among the wide range of possible ML algorithms, what is the optimal model for predicting the target variable? What are the hyper-parameters of such a model? Given the vast array of possible answers to these questions, in recent years, there has been growing interest in machine learning (autoML). In order to address these problems, this paper introduces a method based on Tree-based Pipeline Optimization Tool. Genetic Programming (GP) is used to evolve the sequence of operators in the pipeline as well as the parameters of each operator such as the number of trees in a random forest, the number of features pairs to be selected when selecting features, etc. To explore autoML and assess its performance within named entity recognition, we have chosen a problem that has been the subject of a previous extensive study. TPOT was able to scan the model space and create pipelines that surpassed the state-of-the-art performance of the earlier systems.

In this paper, an approach based on the tree-based pipeline optimization tool (TPOT) is applied. This approach uses tree-based genetic programming for finding the model and its hyperparameters, which more precisely predicts the class of Arabic named entities from a text coming from social media. The model structure and parameters are refined to get the best performance from the machine's learning pipeline. Our approach surpasses a strong baseline performance. It achieves a state-of-the-art in the NER task. Our contributions are as follows:

- Demonstrate the effect of TPOT using word representations and embedding on the performance of NER systems;
- find the model and its hyperparameters appropriate for a given dataset;
- Provide comparable NER performance to other systems.

We are planning to release this system publicly in Github[1].

Currently, the calculation cost of numerous and varied models, as well as hyperparameters, are considerably less than the cost of opportunity of a scientist's time. This enables scientists and researchers to make better use of their time to optimize model performance.

One of the main limitations of TPOT is that it may generate an arbitrary/invalid ML pipelines, i.e., it may generate an ML pipeline without solving a classification problem, as no constraints exist within the type of components can be combined. For example, TPOT may create a pipeline with no classification algorithm [34]. This also results in a waste of computational resources, as they are identified as invalid and are assigned a very low suitability value when evaluating pipelines.

The principal structure of this paper is as follows. Section 2 provides an overview of the Arabic language and the challenges related to the recognition of Arabic-named entities. Section 3 reviews previous work on the TPOT and its implementation. Section 4 describes our approach. Section 5 presents the evaluation parameters and NER features used. Section 6, the experimental results are reported. Section 7 draws a conclusion.

2. BACKGROUND

2.1 Evolution of NER

At the Sixth Message Understanding Conference (MUC-6) [47], the term "Named Entity" (NO) was first utilized as a task to identify the names of organizations, persons, and geographic locations in the text, along with expressions of currency, time, and percentage. Since MUC-6, interest in NER has increased, and many scientific events (e.g., CoNLL03 [18], ACE [17], IREX [16] and TREC Entity Track [8]) are focusing a great deal of effort on this topic.

2.2 Arabic Language Aspects and Challenges

"Arabic is a language marked by a rich morphology and a difficult syntax." [4]. It is divided into three main types: Classical Arabic, the language of the Holy Quran, which has been used for over 1500 years. Modern Standard Arabic, as one of the six official languages of the United

[1] https://github.com/

Nations, while most of the research of the Arabic NLP focuses on colloquial Arabic as the spoken Arabic language. It is irregular and varies from country to country and region to region.

In comparison to the English NER, the following are some examples of challenges for the Arab NER [2]:

- **Lack of capitalization**: The capitalization in foreign languages is a powerful indicator of the named entity. But in Arabic, proper nouns are not capitalized, which makes the identification of ENs more difficult;
- **Noun confusion**: Certain words may be proper nouns, nouns, or adjectives. For example, jamiyolap[2] = “جميلة”, which signifies "beautiful" can be either a proper noun or an adjective. Another example, jamAl = “جمال,” which signifies "beauty," would be a noun, but can be a common noun or proper noun ;
- **Agglutination**: Due to the agglutinating nature of Arabic, a named entity (NE) can be attached to different cliques. A morphological analysis pre-processing step must be carried out to identify and categorize these entities. This feature makes the task of the Arabic NER more difficult ;
- **Optional short vowels:** short (diacritical) vowels are facultative in Arabic. At present, the majority of written MSA texts do not contain diacritics, which leads to a considerable level of ambiguity since a single undiacritized word refers to different terms or meanings. Such ambiguity can often be resolved through the use of contextual information [10].

Additionally to the challenges mentioned above, for Arabic NER generally when compared to Latin-based languages, Dialect Arabic NER is confronted with further challenges:

- Insufficient labeled data for the supervised Dialect Arabic NER ;
- Absence of standard spelling or language academics [24]: In contrast to the MSA, many forms of the same word in DA can be rewritten, e.g., mAtEyT$ = ”ما تعيطش,” mtEyt$ = ”متعيتش ,” that means "don't cry," are all acceptable forms since there is no single standard ;
- Absence of comprehensive gazetteers: an issue confronting any system of NERs for any language that deals with NERs in social media text, as by definition these media have an omnipresent presence of very productive names illustrated by the use of pseudonyms; thus the PERSON class in social media NERs will still have a problem of coverage ;
- The application of NLP tools developed for MSA to DA yields significantly lower performance, thus requiring resources and tools that specifically address DA [23].

3. RELATED WORK

Basically, several works in the literature presented the contribution of the automation of the ML Workflow using TPOT to solve different problems. As far as we know, we did not find a paper that benchmarks its use in an NLP problem.

The utilization of TPOT has grown significantly in the field of neuroscience.

In fact, [30], [31], [45] apply TPOT to real RNA-Seq data taken from a major depressive disorder study. Regardless of the precedent study that identified a meaningful correlation with the severity of depression in the enrichment scores of two modules, in an automated manner, the TPOT corroborates that one of the modules is highly predictive of the clinical diagnosis of each individual.

In addition to this, [34], [36], [37], apply a tree-based pipeline optimization tool (TPOT) and prove its efficiency over a series of simulated and real-world baseline data (predict prostate cancer, predict angiographic diagnoses of coronary artery disease). Specifically, they show that TPOT is capable of designing machine learning pipelines that provide a meaningful enhancement to a basic machine learning analysis without requiring little or no prior data or knowledge from the user. They further address TPOT's trend to build overly complicated pipelines by incorporating Pareto optimization, making pipelines compact without sacrificing classification accuracy. Regarding [11], explore the application of TPOT by using an autoML approach to structural neuroimaging data. They assessed its efficacy in predicting chronological age using

[2] We use the Buckwalter encoding system to render Arabic. http://www.qamus.org/transliteration.htm

structural brain data. They demonstrated that the Ageing is a major cause of structural variability in the brain. Also [38], provide a focused case study addressing AutoML considerations for the use of the TPOT in metabolic profiling of exposure to metformin in a biobank cohort.

TPOT has also shown its effectiveness in the area of Radar signals, as found in [54]. They propose a method based on TPOT and LIME. The pipeline structure and associated parameters are optimized using the genetic algorithm to get the pipeline having an optimal performance over the entire pipeline. Next, the interpreter checks the reliability of the classification model and provides the interpretation of the weighting of the characteristics for the radar signals. Also, [51] presents a proposed LPI radar waveform recognition system using a convolutional neural network and a TPOT to solve the problem of extracting the characteristics of the LPI radar signal and the low recognition ratio of many types of radar waveforms.

4. METHODS

4.1 *Pipeline Optimization*

Currently, the majority of machine learning necessitates engineers to build features, select appropriate features, and choose an appropriate classifier for a given problem based on experience. There are problems in optimizing the classifier's hyperparameters and a series of process problems that engineers must solve based on experience. And for the different recognition objects, the Pipeline parameters are fixed differently. Pipeline tuning is based on TPOT, which is based on genetic programming. It can automate the selection of hyperparameters, utilize a diverse range of modeling algorithms, and finally optimize for an optimal performance pipeline.

4.2 *Tree-based Pipeline*

In TPOT, the data processing flow is illustrated in Figure 1. The tool includes all the processes in which data processing has many operators. These operators are incorporated as genetic programming primitives to build the GP tree, to carry out the machine's automatic learning.

- **Decomposition**: Randomized PCA, representing a variant of principal component analysis using randomized SVD, is applied to decompose the dimensionality reduction [32].

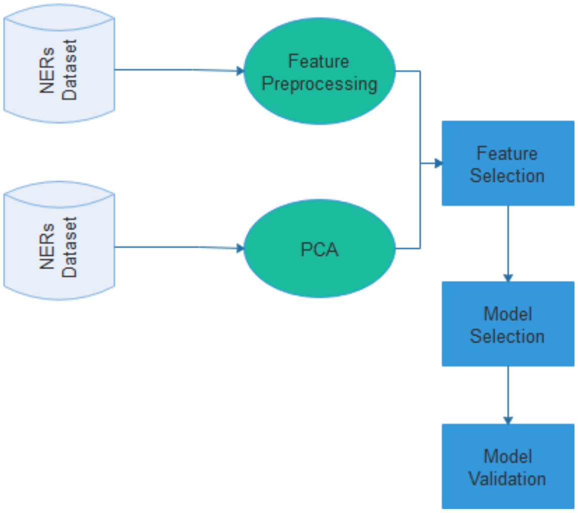

Figure 1: Tree-based pipeline automatic machine learning flow chart

- **Preprocessors**: The TPOT tool operator scales the characteristics utilizing the sample mean and variance (StandardScaler), scales the features using the sample median and interquartile range (RobustScaler), and produces the interacting features using the polynomial combination of numerical characteristics (Polynomial Features). In cases where the number of features is 4, and the degree is 2, the Polynomial Features conversion formula may be given as follows:

$$\sum_{k=1}^{15} x'_k = x_i * x_j$$

Where the values of i and j are $i \leq j$, j = 0, 1, 2, 3, 4. When the values of i and j are 0, the values of x_i and x_j are 1.

- **Feature Selection**: It allows constructing a subset of features from the original feature set by removing irrelevant or redundant features. This process aims to make the model simpler, avoiding over-fitting, and enhancing the model's performance. The chosen features are typically divergent and strongly correlated with object values. According to [14], a typical feature selection process involves four basic steps (see Figure 2), as follows:

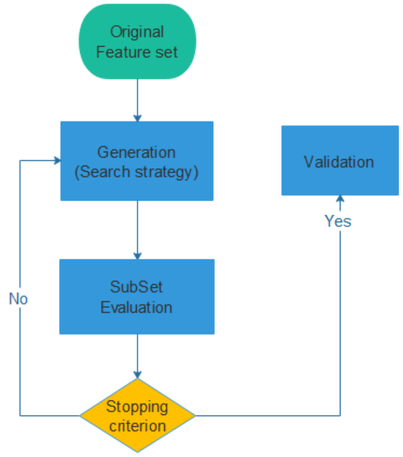

Figure 2: The iterative process of feature selection. A subset of features is selected, based on a search strategy, and then evaluated. Then, a validation procedure is implemented to determine whether the subset is valid. The above steps are repeated until the stop criterion is satisfied.

This tool includes feature selection operators like Select KBest, Select Percentile, and Variance Threshold. Select KBest signifies selecting the first k features, Select Percentile signifies selecting the first k percentiles of the features, and Variance Threshold indicates that a variance threshold is defined. The feature that does not achieve the variance threshold will be rejected. SelectKBest may be combined with chi-square tests and mutual information to select features.

The formula for the chi-square test is expressed as follows:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Where p(x, y) is the joint distribution function of X and Y, and p(x) and p(y) are the marginal probability density functions of X and Y.

Mutual information is defined as follows:

$$x^2 = \sum_{i=1}^{k} \frac{(A_i - E_i)^2}{E_i}$$

Where A_i is the observed frequency of the i-th value of feature A, and E_i is the expected frequency of the i-th value of feature A.

- **Models**: The tool is intended for supervised learning with an integrated decision tree classifier, a random forest classifier, a gradient boosting classifier [49], a support vector machine, a logistic regression, and k-nearest neighbor classifier.

All tree-based pipelines begin with the named entity's dataset that serves as the leaves of the tree. There are four main types of processing carried out by the tree nodes: pre-processing, decomposition, feature selection, and model selection, followed by data passing to the downstream node. When several copies of the data set being processed exist, they can be combined into a single data set through a data set combination operator [34]. The data set is then processed by this node. Three variables were used in the pipeline optimization to evaluate every sample [35] that was successfully optimized for the data set. The variable "class" is the real tag of each sample and is utilized to assess the accuracy of each pipeline. The variable "guess" refers to the last estimate for each sample in the pipeline. The "group" variable specifies if each sample is to be utilized for internal training or testing.

4.3 *Genetic Programming*

To generate and optimize these tree-based pipelines automatically, we use the famous evolutionary computing technique known as genetic programming that is integrated into the Python DEAP package [21]. In this article, the GP algorithm is based on the standard program of the evolutionary algorithm, and the appropriate settings are presented in Table 1.

Table 1: Genetic Programming Setting.

GP Parameter	Value
Population Size	20
Generations	10
Per-individual Mutation Rate	90%
Per-individual Crossover Rate	10%
Selection	10% Elitism, Rest 3-way Tournament (2-way Parsimony)
Mutation	Point, Insert, Shrink 1/3 Chance of Each
Unique Replicate Runs	30

- **Population initialization**: Once the parameters are established at empirical values, the system randomly creates a fixed number of tree pipelines at the start of the evolutionary process to form the main population in genetic programming.

- **Individual Evaluation**: These pipelines are further evaluated individually according to the accuracy of their classification.

- **Individual Selection**: in order to generate the next generation population, A copy of the process individual with the highest level of fitness is first created and placed in the offspring population until these elite individuals make up 10% of the population. Three pipelines are randomly selected from the existing population, and the winning pipelines are placed into the next generation by participating in the tournament. In this process, the worst-performing pipelines are excluded, then the least complex pipeline is chosen among the last two remaining individuals, and the replication is selected into the next generation of the population. The selection process is repeated until the remaining 90% of the offspring are filled.

- **Crossover operator**: Following the creation of the next generation of the population, then two pipelines are randomly selected, and a crossover operator is employed to copy the percentage of the pipeline. They separate at a random point in the tree structure and then exchange their contents with each other.

- **Mutation Operator**: The other uninfluenced ones were randomly changed by shrinkage, insertion, and uniform mutation.

Once the crossover and mutation operations are terminated, the preceding generation of pipelines is completely removed, and the evaluation is repeated with the fixed algebra - the selection - crossover - mutation process. In this manner, the GP algorithm continuously changes the pipelines, adds new operating nodes, enhances fitness, and removes useless or impactful operating nodes.

5. EVALUATION

5.1. Named entity features

The key features of the NER task are determined based on the various possible combinations of available word and tag contexts. To build a NER system based on TPOT, we use the following features:

- **Lexical Features (LEX)**: character n-gram features, the leading and trailing character bigrams (L2), trigrams (L3), and quadrigrams (L4). And the stem, the root of the word as described in [2].

- **Contextual Features (CTX)**: undiacritized words surrounding a context window = ±1;(W-1,W0,W1);

- **Gazetteer (GAZ)**: A binary characteristic indicating whether the word exists in an individual Gaz. In our system, the GAZ is a combination of (i) ANERGaz[3]: As provided by [52], which contains 2183 LOC, 402 ORG, and 2308 PER; and (ii) WikiGaz: Wikipedia, large Gazetteers provided by [13], which lists 50141 LOC, 17092 ORG, and 65557 PER.

- **Morphological Features (MORPH)**: Such features are generated by the MADAMIRA[4] tool [40]. Five morphological features have been selected to be used in this work:
  - o **Aspect**: refers to the aspect of an Arabic verb. There are four possible values: Command, Imperfect, Perfective, Not applicable. Since none of the NEs can be verbal, however, we apply this feature as a binary feature specifying whether a word is marked for Aspect or not;
  - o **Gender**: The nominal gender. It has three values: Female, Male, Not applicable;
  - o **Person**: It indicates person's information. Feasible values are 1st, 2nd, 3rd, Not applicable. As for the aspect, we use it as a binary feature indicating whether a word is tagged for person or not;
  - o **Proclitic2**: The proclitic conjunction. The tool we used generates nine values for this feature: No proclitic, Not applicable, Conjunction fa, Connective particle fa, Response conditional fa, Subordinating conjunction fa, Conjunction wa, Particle wa, Subordinating conjunction wa;
  - o **Voice**: The verb voice. It has the following values for this feature: Active, Passive, Not applicable, Indefinite.

[3] http://users.dsic.upv.es/~ybenajiba/resources/

[4] ttp://innovation.columbia.edu/technologies/cu14012_arabic-language-disambiguation-for-natural-language-processing-applications

- **Part of speech (POS)**: We use POS tags generated from the MADAMIRA tool;
- **Word2vec (W2V)**: Word representations derived from untagged text have successfully proven valuable for numerous NLP tasks, especially for part-of-speech (POS) tagging [26], named entity recognition [42], chunking [27], and parsing [9]. In large corpus, names appear in regular contexts that will be profitable for most sequence tagging tasks: such as NER. So that we could initialize our word vectors with pre-trained word embedding. [46] Demonstrate that the use of embedded words may encode morphological information and can add additional information to the embedding of character-based words.

In our work, we used AraVec[5] which proposed by [42] and is a pre-trained open-source word embedding project that aims to provide the Arabic NLP research community with powerful and freely usable word embedding models. The latter pre-trained by a total of more than 1,476,715 tokens in Twitter with the Skip-Gram algorithm, and we use an embedding dimension of 300 vectors.

- **Word2vec Cluster IDs (W2V Cluster)**: Word2vec[6] is an algorithm for learning embedding utilizing a neural network model proposed by [50]. Embeddings are expressed through a set of latent variables, in which every word is given by a specific instantiation within these variables. In our system, we apply K-means clustering on word vectors and use cluster IDs as characteristics.
- **Brown Clustering IDs (BC ID)**: Brown clustering[7] Provide by [39] is a hierarchical clustering approach that maximizes the mutual information of word bigrams. Word representations, especially Brown Clustering, have been shown to improve the performance of NER system when added as a feature [27]. In this work, we use Brown Clustering (BC) variable IDs as features, resulting in the following set of features. This feature is based on the assumption that semantically similar words will be clustered in the same cluster and will have a shared prefix ;

5.2. *Evaluation metrics*

In order to evaluate the proposed approach, CoNLL’s NER standard evaluation script[8] was used. As discussed extensively in [28], CoNLL’s evaluation methods are one of the most aggressive methods as no partial credit is given for a partially extracted named entity. Outcomes after the CoNLL evaluation script is run are given for each NER class in terms of accuracy, recall and f-score [44].

- **True Positive (TP)**: Entities that are recognized by NER and correspond to the truth on the ground.
- **False Positive (FP):** Entities that are recognized by NER but do not correspond to the truth on the ground.
- **False Negative (FN):** entities annotated in the basic truth that are not recognized by NER.
- **Accuracy** measures the ability of a NER system to present only the correct entities, and recall measures the ability of a NER system to recognize all entities in a corpus.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F-score is the harmonic mean of precision and recall, and the balanced F-score is the most used:

$$\text{Fscore} = 2 * \frac{\text{Precision} \ * \ \text{Recall}}{\text{Precision} \ + \text{Recall}}$$

5.3. *Pre-processing*

Normalization is carried out after removing punctuation, diacritics, and stop words from the text. We employed diacritics suppression, punctuation suppression and Arabic normalization provided by the AraNLP[9] library for the pre-processing of texts. AraNLP developed by [6]. The author in [29] Described stop-words as "words that have no significant semantic relationship with the context in which they exist" [5]. We have established a list of the most frequent stop words

[5] https://github.com/bakrianoo/aravec
[6] https://code.google.com/p/word2vec/
[7] https://github.com/percyliang/brown-cluster
[8] https://github.com/sighsmile/conlleval/blob/master/conlleval.py
[9] https://sites.google.com/site/mahajalthobaiti/resources

that have occurred in corpora. It contains prepositions, conjunctions, punctuation marks, and numbers. Examples of Arabic stop words are: "(In) "(في)," (who) الذي, and (he/she)" هو\هي.

After we used the word stemming in Arabic, that means the process of deleting all prefixes and suffixes from a word in order to generate the stem or root. It is a process of converting plurals into singulars or deriving a verb from the gerund form. Other possibilities include deriving the root from model words. This process of derivation is important for classifiers and index builders/researchers because it reduces dependency on particular word forms and reduces the potential size of vocabularies, which otherwise might have to contain all possible forms. In our work, we used the ISRI Arabic Stemmer [48].

6. EXPERIMENTAL AND RESULTS

We performed many experiments that represent the combination of several features over our model in order to understand their impact on the Arab NER system on NEWS Dataset. We outlined the findings of these experiments.

6.1 *Baseline*

In this section, we evaluate our approach for the task of Arabic NER. We tested our approach described in Sect. 4 on three different datasets and compared it with various state-of-the-art approaches and baselines. More precisely, we compared our approach to both FARASA [1] and MADAMIRA [40], which are currently considered the state-of-the-art Arabic NER tools as based on recent evaluations.

6.2 *NEWS dataset*

The first dataset we evaluated our model on is the NEWS dataset, as an annotated corpus for the Arabic NER task developed by [12]. The NEWS dataset contains 292 sentences that have been extracted from the RSS feed with the Arabic (Egypt) version of news.google.com as of October 6, 2012. The corpus includes news from various sources and covers international as well as local news related to politics, finance, health, sports, entertainment, and technology.

Table 2: NER results for the NEWS dataset

Features	Location	Organization	Person	Overall
	F1	F1	F1	Avg. F1
MADAMIRA	38.8	12.6	29.4	28.4
FARASA	73.1	42.1	69.5	63.9
CTX	74.00	60.34	79.55	72.50
GAZ	74.63	61.80	83.50	75.30
LEX	74.29	59.83	82.57	73.93
POS	73.93	60.94	84.82	75.30
BC	80.36	62.66	87.69	78.87
MORPH	81.25	63.60	88.60	79.62
W2V Cluster	81.78	64.04	90.36	80.99
W2V	**82.73**	**66.70**	**90.91**	**81.58**

Bold values indicate the best performance for the various experiments

From Table 2, we see from the NEWS dataset that FARASA has an average high F-measure of 63.9 compared to MADAMIRA. In contrast, MADAMIRA has a lower mean F-measure when compared to our model. MADAMIRA has the weakest average F-measure of 28.4. The model, which was based on contextual features, performs significantly better when compared to FARASA. The model that obtains the highest mean F-measure for all classes is the approach using the word2vec feature, with an average F-measure of 81.58.

The experiments show that the tremendous increase in the overall performance of the system has been observed with the use of TPOT, which gives us an improvement of 7.42 points in the F1 score. We compare our system with two other models. The highest score reported on this task was achieved by [25]. Their system uses a deep co-learning approach using semi-labeled and BI-LSTM-CRF on the top of the system. They scored 74.1 points in the F1 score. The same NEWS dataset was used by [12] to test their model, which integrates cross-language features and English knowledge bases via multilingual links. They scored 63.9 points in F1.

Table 3: Comparison of our system with two other models on NEWS Dataset

Systems	Location	Organization	Person	Overall
	F1	F1	F1	Avg. F1
Darwish 2013 [12]	73.1	42.1	69.5	63.9
Helwe 2019 [25]	81.6	52.7	82.4	74.1
Our system	**82.7**	**66.7**	**90.9**	**81.5**

As shown in Figure 3, it summarizes the results for the overall state of the art system.

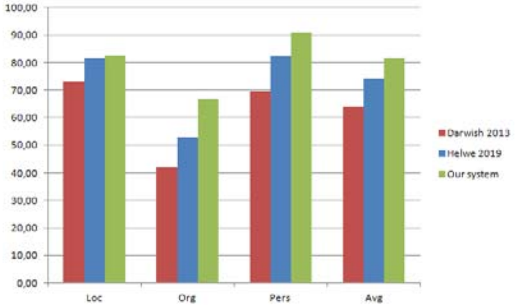

Figure 3: NEWS Test Set Results

6.3 *DA-EGY dataset*

The second dataset is DA-EGY Dataset, which is an annotated data extracted from a set of weblogs that are manually recognized by LDC as Egyptian dialect and which contain almost 40k tokens. The data was tagged by a native Arabic speaker who followed the Linguistic Data Consortium's guidelines for tagging. We utilize the similar 80/20 train/test 5 cross-validation split given in [56].

We carried out several experiments that represent the combination of a different set of features on our model intending to understand their impact on the Arab NER system on the DA-EGY dataset. Table 4 presents the results of these experiments.

The outcome provided is the average of 5-fold cross-validation. Following the proposal in [56], we exclude the ORG class due to the fact that there are less than 0.05% ORG cases in the annotated data, not representing good training data for the system.

Table 4: NER results for the DA-EGY dataset

Features	Location	Person	Overall
	F1	F1	Avg. F1
MADAMIRA	56.58	36.18	62.96
FARASA	61.54	37.68	64.25
CTX	65.5	38.78	66.76
GAZ	73.56	43.58	68.25
LEX	74.23	45.89	68.74
POS	78.45	49.15	70.85
BC	81.01	56.45	72.15
MORPH	86.71	59.56	75.48
W2V Cluster	90.37	61.49	79.82
W2V	**98.10**	**62.48**	**80.45**

Bold values indicate the best performance for the various experiments

Our system surpasses the state-of-the-art scores by 3.33% by using word2vec feature and 2.7% without using word2vec. By using the features of word2vec, we find that the model, as shown in Table 5, generates highly comparable results (80.45%) to the best performance of 77.12% proposed by [55]. In Figure 4, we summarize the results for the overall state of the art system in the literature.

Table 5: Comparison of our system with two other models on DA-EGY Dataset

Systems	Location	Person	Overall
	F1	F1	Avg. F1
Zirikly 2014 [56]	91.43	49.18	70.31
Zirikly 2015 [55]	96.77	57.47	77.12
Our system	**98.10**	**62.48**	**80.45**

As shown in Figure 4, it summarizes the results for the overall state of the art system.

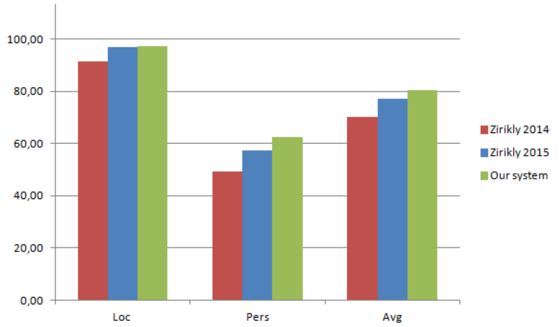

Figure 4: DA-EGY Test Set Results

6.4 *TWEETS dataset*

The third and final dataset we used for evaluation is the TWEETS dataset. We use the split of the training and test data provided in [12], where the training data set comprises 3,646 tweets that were randomly selected and extracted from tweets written between May 3 and May 7, 2012. The tweets were retrieved from Twitter utilizing the lang:ar request. The test data consists of 1,423 tweets that were randomly extracted from tweets written between November 23, 2011, and November 27, 2011. This dataset was similarly tested in [13]. The two datasets are tagged according to the ACE tagging guidelines of the Linguistics Data Consortium. Table 6 gives the statistics.

Table 6: Twitter Evaluation data statistics

	Tokens	PER	LOC	ORG
Twitter Train	55k	788	713	449
Twitter Test	26k	464	587	316

In Table 7, we also find that FARASA scores a high mean F1 of 39.9 in comparison to MADAMIRA on the TWEETS data set, whereas MADAMIRA scores a lower F1 in comparison to our model. Similarly, for the TWEETS dataset, MADAMIRA scores the lowest mean F1 at 24.6. The model with the highest F1 is the approach using the word2vec feature with an average F1 of 65.5. We got the lowest F1 for the model in comparison to the other two datasets because the tweets are very different in nature from the Wikipedia data, which were used to train the model due to errors and misspellings and many tweets were also in Egyptian dialect.

We conclude that our approach using word2vec feature is making a significant improvement in the performance of the Arabic NER task.

Table 7: NER results for the TWEETS dataset

Features	Location	Organization	Person	Overall
	F1	F1	F1	Avg. F1
MADAMIRA	40.3	8.9	18.4	24.6
FARASA	47.5	24.7	39.8	39.9
CTX	52.25	42.27	50	48.65
GAZ	53.77	42.48	50.43	49.40
LEX	68.03	45.54	52.84	56.57
POS	68.40	45.25	52.41	56.50
BC	65.73	43.42	53.20	55.14
MORPH	62.59	44.77	69.80	61.08
W2V Cluster	68.13	40.69	72.17	64.28
W2V	**72.40**	**46.72**	**73.70**	**65.66**

Bold values indicate the best performance for the various experiments

Compared to the results in the literature, the system performs better than the state-of-the-art Arabic NER systems as applied to the TWEETS dataset, as shown in Table 8. Our system surpasses previous systems in terms of F1 for the retrieval of NEs among locations, organizations, and persons from TWEETS with an overall F1 of 65.5%.

The reason for our good results compared to previous Arabic NER systems may be explained by the way our proposed feature set combines word2vec features that produce state-of-the-art results with powerful features such as W2V Cluster feature, POS feature and morphological features. This combination creates a set of optimized features that allow our model to achieve state-of-the-art results.

Table 8: Comparison of our system with other models on TWEETS Dataset

Systems	Location	Organization	Person	Overall
	F1	F1	F1	Avg . F1
Darwish 2013 [12]	47.5	24.7	39.8	39.9
Darwish 2014 [13]	65.5	41.5	48.5	55
Zayed 2015 [53]	-	-	66.75	-
Helwe 2019 [25]	65.3	39.7	61.3	59.2
Zirikly 2015 [55]	61.03	41.8	68.92	59.5
Our system	**72.4**	**46.7**	**73.7**	**65.6**

As shown in Figure 5, it summarizes the results for the overall state of the art system.

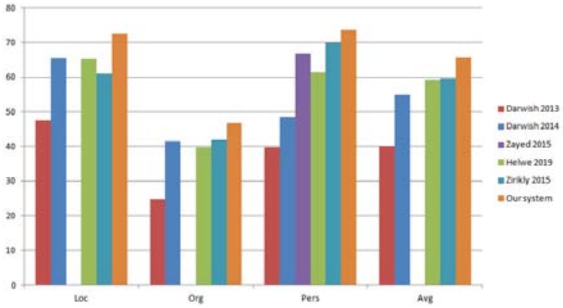

Figure 5: TWEETS Test Set Results

7. DISCCUSION

In this paper, we have demonstrated that TPOT is able to build machine learning pipelines that reach concurrent classification accuracy and discover new pipeline operators such as the selection of synthetic Features that considerably enhance the efficiency of classification.

The most appropriate selection of an ML pipeline to predict such variables of interest (such as the named entity) from given data is determined by the data's statistical characteristics and distribution. In most cases, the machine learning model's selection to be applied to multivariate analysis of named entity data is mostly arbitrary - relying on previous models that "worked" or choosing the model that is newest to the analyst community. In order to consider an alternative approach to selecting a model for a reasonably straightforward problem, in this work, we explored the implementation of an automated analysis technique: TPOT. The TPOT approach is a data-based methodology that is agnostic to the statistical pattern and prepossession of the dataset - seeking to determine the optimally available pipeline for fitting the statistical properties of the underlying dataset, while simultaneously controlling for overfitting and robustness.

Importantly, designing automated pipelines is neither replacing data scientists nor the practitioners of machine learning. Instead, the purpose of the tree-based pipeline optimization tool (TPOT) is to be a "scientific assistant" for exploring the data, discovering new features in the data, and recommending pipelines to the user. Hence, the user can export the pipelines freely and incorporate their field expertise as needed.

8. CONCLUSION

Aiming at resolving the problems of natural language processing tasks, in particular the name entity recognition, we introduce a method relying on Tree-based Pipeline Optimization Tool (TPOT). This method utilizes genetic programming based on the tree-structure to generate the machine learning pipeline, in which the structure and parameters are evolved to obtain optimal performance. This is one step towards the application of machine learning techniques to automate the construction of classification systems for difficult text processing problems. Overall, our results demonstrate that the TPOT approach may be used as a data-driven approach to identify ML models that precisely predict the named entity. Hence, TPOT-produced models could be generalized to unseen datasets and had considerably higher performance.

REFRENCES:

[1] Abdelali, A., Darwish, K., Durrani, N. and Mubarak, H. 2016. Farasa: A Fast and Furious Segmenter for Arabic. 2016, (2016), 11–16. DOI:https://doi.org/10.18653/v1/n16-3003.

[2] Abdul-hamid, A. and Darwish, K. 2010. Simplified Feature Set for Arabic Named Entity Recognition. Proceedings of the 2010 Named Entities Workshop. July (2010), 110–115.

[3] ADIBA MAHJABIN NITU, MD. PALASH UDDIN, PRIYANKA BASAK TUMPA, SABINA YEASMIN, M.I.A. 2019. AN ENHANCED EXTRACTIVE TEXT SUMMARIZATION. Journal of Theoretical and Applied Information Technology. 97, 23 (2019), 3475–3485.

[4] Al-Kharashi, I.A.A.-S. and I.A. 2004. Arabic morphological analysis techniques: A survey and classification. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY. 55, 3 (2004), 189–213.

[5] Alajmi, A. and Saad, E. mostafa 2018. Toward an ARABIC Stop-Words List Generation Toward an ARABIC Stop-Words List Generation. January 2012 (2018).

[6] Althobaiti, M., Kruschwitz, U. and Poesio, M. 2014. AraNLP: A Java-based library for the processing of Arabic text. Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014. August 2015 (2014), 4134–4138.

[7] B. Babych and A. Hartley 2003. Improving

machine translation quality with automatic named entity recognition. in Proc. EAMT.

[8] Balog, K., Serdyukov, P. and De Vries, A.P. 2011. Overview of the TREC 2011 entity track. NIST Special Publication. January (2011).

[9] Bansal, M., Gimpel, K. and Livescu, K. 2014. Tailoring continuous word representations for dependency parsing. 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference. 2, (2014), 809–815. DOI:https://doi.org/10.3115/v1/p14-2131.

[10] Benajiba, Y., Rosso, P. and Diab, M. 2009. Arabic Named Entity Recognition: A Feature-Driven Study. IEEE Transactions on Audio, Speech and Language Processing. 17, 5 (2009), 926–934. DOI:https://doi.org/10.1109/TASL.2009.2019927.

[11] Dafflon, J., Cole, J.H., Turkheimer, F., Leech, R., Harris, M.A., Cox, S.R., Whalley, H.C., McIntosh, A.M. and Hellyer, P.J. 2019. Automated Machine Learning in Brain Predictive Modelling: A data-driven approach to Predict Brain Age from Cortical Anatomical Measures. (2019). DOI:https://doi.org/10.32470/ccn.2019.1080-0.

[12] Darwish, K. 2013. Named entity recognition using cross-lingual resources: Arabic as an example. ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. 1, (2013), 1558–1567.

[13] Darwish, K. and Gao, W. 2014. Simple effective microblog named entity recognition: Arabic as an example. Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014. c (2014), 2513–2517.

[14] Dash, M. and Liu, H. 1997. Feature selection for classification. Intelligent Data Analysis. 1, 3 (1997), 131–156. DOI:https://doi.org/10.3233/IDA-1997-1302.

[15] David Nadeau, S.S. 2007. A survey of named entity recognition. Communications. 30, 1 (2007), 14–30. DOI:https://doi.org/10.1075/li.30.1.03nad.

[16] Demartini, G., Iofciu, T. and De Vries, A.P. 2010. Overview of the INEX 2009 entity ranking track. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 6203 LNCS, (2010), 254–264. DOI:https://doi.org/10.1007/978-3-642-14556-8_26.

[17] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R. 2004. The automatic content extraction (ACE) program tasks, data, and evaluation. Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004. (2004), 837–840.

[18] Erik F. Tjong Kim Sang and Fien De Meulder 2013. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. Pediatric Blood and Cancer. 60, SUPPL.1 (2013). DOI:https://doi.org/10.1002/pbc.24340.

[19] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S. and Yates, A. 2005. Unsupervised named-entity extraction from the Web: An experimental study. Artificial Intelligence. 165, 1 (2005), 91–134. DOI:https://doi.org/10.1016/j.artint.2005.03.001.

[20] Finkel, J.R., Grenager, T. and Manning, C. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. 1995 (2005), 363–370.

[21] Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M. and Gagńe, C. 2012. DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research. 13, (2012), 2171–2175.

[22] Guo, J., Xu, G., Cheng, X. and Li, H. 2009. Named entity recognition in query. Proceedings - 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009. (2009), 267–274. DOI:https://doi.org/10.1145/1571941.1571989.

[23] Habash, N., Diab, M. and Rambow, O. 2012. Conventional orthography for dialectal Arabic. Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012. January (2012), 711–718.

[24] Habash, N., Roth, R., Rambow, O., Eskander, R. and Tomeh, N. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. Proceedings of NAACL-HLT. June (2013), 426–432.

[25] Helwe, C. and Elbassuoni, S. 2019. Arabic

named entity recognition via deep co-learning. Artificial Intelligence Review. 52, 1 (2019), 197–215. DOI:https://doi.org/10.1007/s10462-019-09688-6.

[26] Huang, F., Ahuja, A., Downey, D., Yang, Y., Guo, Y. and Yates, A. 2010. Learning Representations for Weakly Supervised Natural Language Processing Tasks. Dissertation Abstracts International, B: Sciences and Engineering. 70, 8 (2010), 4943. DOI:https://doi.org/10.1162/COLI.

[27] Joseph Turian, Lev Ratinov, Y.B. 2010. Word representations: A simple and general method for semi-supervised learning Joseph. Journal of Pharmacy and Pharmacology. 30, 1 S (2010), 53P-53P. DOI:https://doi.org/10.1111/j.2042-7158.1978.tb10760.x.

[28] Khaled Shaalan 2010. A Survey of Arabic Named Entity Recognition and Classificatio. Dissertation Abstracts International, B: Sciences and Engineering. 70, 8 (2010), 4943. DOI:https://doi.org/10.1162/COLI.

[29] Khosrow, M. 2018. Encyclopedia of Information Science and Technology, Fourth Edition Mehdi Khosrow-Pour Information Resources Management Association, USA Category: Hospitality, Travel, and Tourism M... Encyclopedia of Information Science and Technology, Fourth Edition. December (2018), 4077–4087. DOI:https://doi.org/10.4018/978-1-5225-2255-3.ch353.

[30] Le, T.T., Fu, W. and Moore, J.H. 2020. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics (Oxford, England). 36, 1 (2020), 250–256. DOI:https://doi.org/10.1093/bioinformatics/btz470.

[31] Le, T.T., Fu, W. and Moore, J.H. 2020. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics (Oxford, England). 36, 1 (2020), 250–256. DOI:https://doi.org/10.1093/bioinformatics/btz470.

[32] Martinsson, P.G., Rokhlin, V. and Tygert, M. 2011. A randomized algorithm for the decomposition of matrices. Applied and Computational Harmonic Analysis. 30, 1 (2011), 47–68. DOI:https://doi.org/10.1016/j.acha.2010.02.003.

[33] Mollá, D., van Zaanen, M. and Smith, D. 2006. Named Entity Recognition for Question Answering. Proceedings ALTW 2006. September 2009 (2006), 51–58.

[34] Olson, R.S., Bartley, N., Urbanowicz, R.J. and Moore, J.H. 2016. Evaluation of a tree-based pipeline optimization tool for automating data science. GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference. (2016), 485–492. DOI:https://doi.org/10.1145/2908812.2908918.

[35] Olson, R.S. and Moore, J.H. 2019. TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning. (2019), 151–160. DOI:https://doi.org/10.1007/978-3-030-05318-5_8.

[36] Olson, R.S., Urbanowicz, R.J., Andrews, P.C., Lavender, N.A., Kidd, L.C. and Moore, J.H. 2016. Automating biomedical data science through tree-based pipeline optimization. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 9597, (2016), 123–137. DOI:https://doi.org/10.1007/978-3-319-31204-0_9.

[37] Orlenko, A., Kofink, D., Lyytikäinen, L.P., Nikus, K., Mishra, P., Kuukasjärvi, P., Karhunen, P.J., Kähönen, M., Laurikka, J.O., Lehtimäki, T., Asselbergs, F.W. and Moore, J.H. 2020. Model selection for metabolomics: predicting diagnosis of coronary artery disease using automated machine learning. Bioinformatics (Oxford, England). 36, 6 (2020), 1772–1778. DOI:https://doi.org/10.1093/bioinformatics/btz796.

[38] Orlenko, A., Moore, J.H., Orzechowski, P., Olson, R.S., Cairns, J., Caraballo, P.J., Weinshilboum, R.M., Wang, L. and Breitenstein, M.K. 2018. Considerations for automated machine learning in clinical metabolic profiling: Altered homocysteine plasma concentration associated with metformin exposure. Pacific Symposium on Biocomputing. 0, 212669 (2018), 460–471. DOI:https://doi.org/10.1142/9789813235533_0042.

[39] P.~Brown, V.~Della Pietra, de Souza, P., J.~Lai and R.~Mercer 1992. Class-based n-gram models of natural language. Computational Linguistics. 18, 1950 (1992), 467–479.

[40] Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O. and Roth, R.M. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014. (2014), 1094–1101.

[41] Petkova, D. and Croft, W.B. 2007. Proximity-based Document Representation for. (2007), 731–740.

[42] Ronan Collobert, Jason Weston, Léon Bottou, K.M. and Koray Kavukcuoglu, P. 2017. Natural Language Processing (Almost) from Scratch Ronan. Proceedings - 2017 IEEE 3rd International Conference on Collaboration and Internet Computing, CIC 2017. 2017-Janua, (2017), 328–338. DOI:https://doi.org/10.1109/CIC.2017.00050.

[43] Sengloiluean, K., Arch-Int, N. and Arch-Int, S. 2019. A semantic question classification for question answering system using linked open data approach. Journal of Theoretical and Applied Information Technology. 97, 20 (2019), 2293–2305.

[44] De Sitter, A., Calders, T. and Daelemans, W. 2004. A Formal Framework for Evaluation of Information Extraction. July (2004), 1–12.

[45] Sohn, A., Olson, R.S. and Moore, J.H. 2017. Toward the automated analysis of complex diseases in genome-wide association studies using genetic programming. GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference. (2017), 489–496. DOI:https://doi.org/10.1145/3071178.3071212.

[46] Soricut, R. and Och, F. 2015. Unsupervised morphology induction usingword embeddings. NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference. (2015), 1627–1637. DOI:https://doi.org/10.3115/v1/n15-1186.

[47] Sundheim, B., Road, G., Diego, S., Grishman, R. and York, N. Grishman, Sundheim – Message Understanding Conference.

[48] Taghva, K., Elkhoury, R. and Coombs, J. 2005. Arabic stemming without a root dictionary. International Conference on Information Technology: Coding and Computing, ITCC. 1, (2005), 152–157. DOI:https://doi.org/10.1109/itcc.2005.90.

[49] Tianqi Chen, C.G. 2016. XGBoost: A Scalable Tree Boosting System. Il Friuli medico. 19, 6 (2016).

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, J.D. 2016. Distributed Representations ofWords and Phrases and their Compositionality. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings. (2016), 1389–1399. DOI:https://doi.org/10.18653/v1/d16-1146.

[51] Wan, J., Yu, X. and Guo, Q. 2019. LPI radar waveform recognition based on CNN and TPOT. Symmetry. 11, 5 (2019), 1–15. DOI:https://doi.org/10.3390/sym11050725.

[52] Yassine Benajiba, P.R. 2008. Arabic named entity recognition using conditional random fields. 2008 5th International Conference on Information and Communication Technology, ICoIC7 2008. (2008). DOI:https://doi.org/10.1109/ICoICT.2017.8074647.

[53] Zayed, O.H. and El-Beltagy, S.R. 2015. Named entity recognition of persons' names in Arabic tweets. International Conference Recent Advances in Natural Language Processing, RANLP. 2015-Janua, (2015), 731–738.

[54] Zhang, W., Ge, P., Jin, W. and Guo, J. 2018. Radar Signal Recognition Based on TPOT and LIME. Chinese Control Conference, CCC. 2018-July, (2018), 4158–4163. DOI:https://doi.org/10.23919/ChiCC.2018.8483165.

[55] Zirikly, A. and Diab, M. 2015. Named Entity Recognition for Arabic Social Media. (2015), 176–185. DOI:https://doi.org/10.3115/v1/w15-1524.

[56] Zirikly, A. and Diab, M. 2014. Named Entity Recognition System for Dialectal Arabic. (2014), 78–86. DOI:https://doi.org/10.3115/v1/w14-3610.