Classifiers Fusion for Arabic Named Entity Recognition

Wasim M. Abdulwasea*[1], SherifAbdou*[2], Hassanin Al-Barhamtoshy**[3]
*Information Technology, Faculty of Computers and Information, Cairo University , Giza, Egypt
[1]w.abdulwasea@gmail.com,
[2]s.abdou@fci-cu.edu.eg
**Faculty of Computing & Information Technology, King AbdulazizUniversity, Saudi Arabia
[3]hassanin@kau.edu.sa

Abstract: ***This paper presents a new approach to Arabic Name Entity Recognition (ANER). The introduced approach uses different sets of features that are both language independent and language specific in a discriminative and generative machine learning frameworks namely, conditional random fields (CRF), support vector machines (SVM), Naive Bayes(NB), Decision Tree (DT), SVM for sequence tagging using Hidden Markov Models (SVM^{hmm}), K-nearest neighbors(K-NN), Logistic classifier and the other SVM Weka model called (SMO). Also all these classifiers have been fused together and the fusion configuration provided more accurate ANER than any one of the classifiers when used individually. The proposed approach has been evaluated using two data sets, the first dataset is a recently published corpus called ALTEC Named Entity Corpus for Modern Standard Arabic proposed by the Arabic Language Technology Center (ALTEC), and the second dataset is a standard dataset in Arabic NER called ANERcrop proposed by Benajiba. The proposed approach proved that it outperforms state of art Arabic NER systems for both of the two data sets using the 6-fold evaluation criterion.***

Keywords: Information Retrieval, Name Entity Recognition, Classifiers Fusion

1 INTRODUCTION

The recognition and classification of proper names in text (e.g. persons, locations, and organizations) has been recently considered of major importance in Natural Language Processing (NLP) as it plays a significant role in various types of NLP applications, especially in Information Extraction, Information Retrieval, Machine Translation, Text clustering, Syntactic Parsing/Chunking, Question-Answering, Text to Speech(TXT) and many other applications. The valuable information in text is usually located around proper names, to collect this information it should be found first [1]. The NER task is defined as the identification and classification of Named Entities (NE's) within an open-domain text into predefined types, such as Person, Location and Organization names [2][3].

Recognizing NE’s consists of two tasks, the first one is identifying a possible NE, and the second one is to classify it into one of possible NE types. Identifying a possible NE is a problem known as boundary detection and it involves determining where a NE begins and ends in a document. For example, “Real Madrid” is a football team, but also “Madrid” on its own is a location. The quality of the NER system has a direct impact on the quality of the overall NLP applications that employ NER as an important preprocessing step to enhance the overall performance.

The NER for the Arabic language faces many challenges due to the complicated nature of the Arabic language. Arabic Language is not a case-sensitive language; it has no capital letters unlike the European languages where an NE usually begins with a capital letter. Also Arabic is a high inflectional language; often a single word has more than one affix. Coordinating conjunctions, prepositions, possessive pronouns, and determiners are typically attached to words as prefixes or suffixes. Another challenge for NER in Arabic is the absence of short vowels, or diacritics in most of modern Arabic texts. These diacritics, specially the case ending mark, can disambiguate between a NE and other meanings depending on the context. For example the Arabic NE "Adel" can be an adjective which means “Fair”.

To build NER systems for any language we have three different approaches; the first one depends on the linguistic knowledge, especially human intuition and grammar rules so this approach is called Rule-Based (RB) approach. The second approach depends on statistical techniques called Machine Learning(ML) approach. A thirdapproach combines the previous two approaches by utilizing the output of RB as feature in ML training phase and it iscalled hybrid approach. Since the patterns of errors for the NER taskthat are produced by using different classifiers are usually independent, the combination of different classifiers for that task can lead to more accurate system. This approach is called multiple classifier system (MCS) or classifier fusion and it combines several ML models into an ensemble, using some of linear and non-linear combinations methods. To the best of our knowledge, the MCS approach for the Arabic NER has not been investigated before.This paper presents a new ANER system based on the fusion of multiple classifiers. A set of fusion methods have been investigated such as majority vote,maximum,average, product and stacking.In the following sections of this paper, section 2 includes a literature review for the recent efforts of Arabic NER, section 3 introduces our approach for ANER including four phases: classifiers, feature-sets combinations, fusion methods and model selection. Section 4describes the used evaluation data for Arabic NEs. Section 5 includes the system experimental results and analysis. Section 6 includes the conclusions and some prospects for future work.

2 LITERATURE REVIEW

A. NER Using Individual Classifier

Benajiba and his team[4] developed the system ANERsys 1.0, which uses Maximum Entropy (ME) learning and a set of lexical, contextual and gazetteers features. The authors have built their own linguistic resources of Arabic NE annotated corpus and gazetteers. This system can recognize four types of NEs: Person(PERS), Location(LOC), Organization(ORG) and Miscellaneous(MISC) and achieved an F1-measure of 55.23%. That system had difficulties for detecting NEs that are composed of more than one token or word. An enhanced version ANERsys2.0[5] used two step technique by detecting the NE boundary then classifying the delimited NEs and achieved F1 of 66%. The used features for that system were Part Of Speech (POS) tags, Base Phrase Chunks (BPC), gazetteers and nationality. Changing the probabilistic model from ME to (CRF)[6] improved the results significantly with an F1-measure of 79.2%. Another update of that system used a classifier based on (SVM)[7] and features that are contextual, lexical, morphological, gazetteers, POS-tags, BPC, nationality and the corresponding English capitalization. That system achieved for the ANERcorp an F1 of 80.4%. Both of the language independent and language specific features were confirmed to be effective for ANER[3][8]. A simplified set of features were proposed in an ANER system to recognize three types of NEs: PERS, LOC and ORG[9]. That system considers only surface features which are leading and trailing character n-gram, word position, word length, word unigram probability, the preceding and succeeding words n-gram and character n-gram probability. Evaluation results using the ANERcorp and ACE2005 datasets show that this system outperforms the CRF-based ANER system of[6]. An integration approach was proposed by combining CRF and bootstrapping semi-supervised pattern recognition[2]. The 6-fold evaluation results using the ANERcorp data set show that this system outperforms LingPipe NE recognizer[10]. The ANER system described in [10] use (SVM^{hmm}) classifier with a set of dependent and independent language features. Also they use patterns to ameliorate the ANER task by implementing an automatic pattern extractor framework based on (POS) Information and linguistic filter and that system achieves an overall F-measure of 83,20% evaluated using 10% ANERcorp corpus. In another attempt to develop ANER system [11]theneural network classifier is used and achieves an overall accuracy 92% evaluated using 10% ANERcorp. 01001934100

B. NER using Multiple Classifier fusion

Classifier fusion has received considerable attention in last couple of decades and it’s becoming a well-known pattern recognition field of study.Radu Florian *et al.*[12]presents a classifier-combination experimental framework for NER in which four diverse classifiers (robust linear classifier, ME, transformation-based learning, and HMM) are combined under different conditions. When no gazetteer or other additional training resources are used, the combined system attains improvement for the overall system performance, when compared with the best performing classifier. Wang *et al.*[13]present classifiers ensemble approaches for biomedical NER. Generalized Window, CRF, SVM, and ME classifiers are combined through three different strategies which are meta-learning, stacking and cascade generalization. Their experimental results demonstrate that the classifiers ensemble strategy especially the class-attribute stacking method is a suitable method for biomedical NER and can lead to significant improvement in performance of NER systems.Danesh et al.[14]proposed a novel approach for text classification based on combining three classifiers Rocchio, k-NN and Naïve Bayes classifiers using voting algorithms , OWA operators and Decision Template for the classifiers fusion. The proposed solution achieves a better classification rate with the classification error decreased by15%. Chia-Wei Wu et al.[15]haveexploited two ensemble methods for Chinese NER in order to integrate multiple results generated under different conditions. One method is based on majority vote, while the other is a memory-based approach that integrates ME and CRF classifiers. Their results showed that the memory-based method managed to outperform the individual classifiers. A. Ekbal and S. Saha[16] describe a system that uses genetic algorithms to find an optimal classifier ensemble for Bengali NER. The system selects from a set of 19 maximum entropy classifiers. They evaluated the ensemble classifiers on Bengali, Hindi, Telugu and English datasets, and report F-score improvements over the best individual classifiers of %5.6, %1.9, %5.7 and %12.8, for each language respectively. A closely work follows Ekbal system by B. Desmet and V. Hoste[17]who investigated if a similar system can successfully be applied for Dutch NER They constructed the best classifier ensemble from a set of three different classification frameworks, namely memory-based learning, CRF and SVM. Their experiments yielded a classifier ensemble that outperformed the best individual classifier by 0.67% (F-score), a small but statistically significant margin. Experimental results also showed that ensemble classifiers, from different frameworks, can provide better generalization. To the best of our knowledge, there is no Arabic NER system based on classifiers fusion.

3 THE PROPOSED SOLUTION FOR ANER

A. NER Features

Features selection plays a crucial role in the NER systems, their quality is essential to achieve high performance. There is no method for automatic selection of given feature sets. We used several combinations of features to extract the most useful and effective features for the ANER task.

The used features in our system are:

1) Context (CXT): is an automatically generated feature that accounts for the different contexts in which NEs appear in the training data. The context is defined as a window of +/-n tokens from the NEs of interest. Based on data observation, we found that surrounding words hold effective information for the recognition of NEs so we used context window size of -1/+1 which achieved better performance than using bigger window.

2) Part of Speech (POS) tagging: The POS tags of the current word and the surrounding words are considered an important feature for NER. We used the Stanford POS9 tagger [18] to provide the likely POS tag for each word. Also we tried the AMIRA-2.1 POS tagger.

3) Contains Digit (CD): is a binary feature where the feature is assigned to one if the current word contains digit(s).

4) Determinant (DT): Most Arabic names accept the entry of Lam definition (ال), this feature can be used to distinguish the Arabic names from non-NEs. Word such as (ذهب) which is a verb becomes a name (الذهب) after adding (ال) to the beginning of the word. Also, it is helpful in recognition of many NE classes as many Arabic names of organization such as: المنظمةالعربية,الشركةالمصرية and person last name start with this determinant such as: الفاربي,الايوبي,عبدالرحمنالابنودي

5) Word Length (WL): is a binary valued feature. This feature checks whether the number of characters in the current word is less than a threshold value, that was set to three in our system. This feature is defined based on the data observation that very short words are rarely NEs.

6) Gazetteers (GAZ): Using Gazetteers are very useful to enhance the performance of the NER system due to the limited amount of training material [1].The use of gazetteers in ANER systems has been investigated in [4] and its advantages were highlighted. This feature is binary valued and indicates the presence of the word in the gazetteer's list. We used the ANERGazet5 (GAZ) and boosted them with NEs extracted from Arabic Wikipedia (AWP).Forthe*Location gazetteer* we started with a list that has been enriched with names of places such as: sports stadiums, museums, Arabic and international libraries, hotels, Arabic newspapers, famous blogs, news agencies names… etc. Currently we have 3400 location names in our list. For the *Person gazetteer*we added a list of famous people names in many fields such as politics, science, literature, philosophy and sport, along with a number of foreign names translated to Arabic. Our current person list includes 8480 names. For the *Organization gazetteer* we added the names of Arabic organizations, international organizations along with international, Arabic and multinational companies. We gathered 2141 organization names.

7) Stop Word (SW): a Binary feature set to 1 if the current word appears in the SW list. Our SW list consists of 10350 words including prepositions, demonstrative pronouns, identifiers, logical connectors with all forms.

8) NE Prefix Word List (MFI): Word prefixes are helpful in recognition of NEs. Based on data observations NEs share some common prefix strings. We generated a list for each NE class (LOC, ORG, PERS, MISC) from the training corpus containing the most frequent indicators (MFI)or most frequent word that precede NE which can be the first part of a composed NE. For example: MFI for person (بن, ابو, الشيخ, السيد, د.), MFI for location (مدينة, نادي, شارع) and MFI for organization (حزب, منظمة, مجلس, اتحاد). We check if a word from the list precedes the current word and set the feature to 1 if it applies.

9) Previous Word (PW):PW hold an effective information that indicates if the next word is NE or not. The NEs accept that the previous word to be a preposition or an appeal (حرفنداء) (PW (GAR, NEDA)), while the verbs don’t accept it, and there are words that cannot come before the name (NOT), such as (سوف, قد, سـ).

10) Part of Speech Surrounding Information (POSS):POS information of the current and/or surrounding word(s) are very helpful for NER. Based on the Esnad (إسناد) rule, we found that the true names are usually preceded or followed by verbs [19]. If the POS of the current word is NNP and there is a VB/VBD/VBG/VBN/VBP word within window +1/-1 tokens to the left or to the right of the word, we assign the value of that feature to be 1. Also, adjective can’t precede a name or come between two names in the Arabic language. The feature is set to 1 if the adjective tags (jj /DTJJ/ADJ) didn't not come before NNP or between two NNP.

11) Character-Based Feature (CH): Some of words or part of words can be very helpful in the process of NE detection especially for proper names. Using leading and trailing character n-grams in words such as : “عبد” “Abed” which is a very frequent prefix in Arabic person names and appears in the beginning of a proper name also the word "بن" "Ben" that appears often in the middle of a composed Arabic proper name [2]. Another character based feature is the تنوين “◌ٍ◌ٌ◌ً” that appears on the last character of NEs. This feature is useful with diacretized data.
12) Nationality Feature (NAT):Wards in sentences are checked against a manually created list of nationalities. This feature is useful for detecting person NEs as they are almost preceded or followed by a nationality such as: الرئيس الامريكي بوش, حسن المصري.
13) Previous Word, Next Word (PWNW) and Current Word (CW): We used the previous word, next word and the world itself as features.
14) Base Phrase Chunks (BPC): This feature represents the Base Phrase Chunks (atomic parts) of a sentence. The AMIRA-2.110[20] tool was used to extract this feature.
15) Morphological features: The MADA tool is used to extract 14different morphological features.These feature are *Aspect, Case, Gender, Mood, Number, Person, State, Voice, Capital letters, enclitic definitions and proclitic definitions* [21] .The last two features are used to detect the prefix and suffix of a word and to exactly specify the clitics that are present. These features are organized according to the possible location of the clitic in the word. The proclitic or enclitic number refers to the location of the clitic, according to [PRC3 [PRC2 [PRC1 [PRC0 BASEWORD ENC0]]]].

B. Classifiers

The first step in building fusion system is the designing of individual base classifiers or model, which will participate in the final decision-making. In this phase the diversity of the models must be taken in consideration to guarantee the improvement in system performance. In this work nine different classifiers have been selected which are: CRF, YamCha–SVM model, Naive Bayes(NB), Bayes Net(BN), Decision Trees (DT), SVM for sequence tagging using HMM (SVM^{HMM})[22], K Nearest Neighbor (K-NN) with k=3 , Logistic classifier and the other WEKA SVM model called (SMO). We chose these classifiers because they proved to be the most effective learners for many NLP applications and the NER task in particular. Our selection meets the essential requirements for MCS which are the diversity to provide different perspectives on the problem, accuracy to avoid random guessing and efficiency to be applied without excessively penalizing runtime.

We used the following tool to build our base models:

1) *CRF++:* A free open source toolkit, for learning CRF models in order to segment and annotate sequences of data. The toolkit is efficient in training and testing and can produce n-best outputs. It can be utilized in developing many NLP components and can handle large feature sets.
2) *Weka***:** A collection of ML algorithms developed for data mining tasks. We use this tool to train six of our models which areDT(we used the J48 implantation), Naïve Bayes(NB) and Bayes net (BN), K-NN with k=3, Logistic classifier and SMO. The Weka toolhas been successfully used to develop DT classifier as part of a hybrid Arabic NER system [23].
3) *YamCha***:** it’s a free open source toolkit for learning SVM models. This toolkit is generic, customizable, efficient, and has an open source text chunker. It has been utilized to develop NLP tasks such as NER, POS tagging, base-NP chunking, text chunking, and partial chunking. YamCha performs well as a chunker and is capable of handling large sets of features. Moreover, it allows for redefining feature parameters (window-size) and parsing-direction (forward/backward), and applies algorithms to multi-class problems (pair wise/one vs rest).
4) *SVM^{hmm}*: is an implementation of structural SVMs for sequence tagging e.g. POS, NER, motif finding. Using the training algorithm described in[18][24] and the new algorithm of SVM^{struct} V3.10[25]. SVM^{hmm} discriminatively trains models that are isomorphic to kth-order HMM model using the Structural SVM formulation. This tool can easily handle tagging problems with millions of words and millions of features, can train higher order models with arbitrary length dependencies for both the transitions and the emissions, includes an optional beam search for fast approximate training and prediction. The SVM^{hmm} has been previously used to train and test a model for Arabic NER [10].

C. *Fusion Methods*

Once the individual classifiers have been designed and implemented, the next step in MCS involves the combination of results obtained through each individual classifier. Methods for fusing multiple classifiers can be classified according to the type of information produced by the individual classifiers into three levels with three types of fusion functions [26]. For a given R classifiers (Cj , *j = 1, ...,R*) fusion methods can be classified according to: **1)** a classifier only outputs a unique label for each input pattern called the abstract level output, where each classifier assigns a label θ_j to a given input *x*. Therefore, the classifier fusion function involves the assignment of a definitive label (θ_{MCS}) to *x*. **2)** the rank level output where each classifier outputs a list of possible classes, with ranking for each input pattern, for each input *x* each classifier produces an integer $RANK_j$, *j = 1, ...,R*. Each element within this array corresponds to one of the output classes. The array is usually sorted descending. Therefore, the task of fusion is to produce an integer array ($RANK_{MCS}$) which ranks output classes according to the given input x. **3)** the measurement level:each classifier Cj produces a real vector of the form $Sj = [sj1, \ldots, sjc]$, where, sji denotes the belief value that classifier j has that x belongs to class i. Therefore, the function of the classifier fusion is to build another real vector (Y_{MCS}) to denote its confidence of the input belonging to each output class. These classifiers are also known as probabilistic classifiers. A multiple classifier can be constructed either in a parallel, cascading or combined topology. The selection of a proper topology depends on the type of problem at hand. The main disadvantage of cascading is the inability of later classifiers to correct mistakes made by earlier classifiers.

In our system we investigated the following methods for combining classifiers outputs:

1) Maximum (MAX): The classifier output with the highest value or confidence is chosen as the output of the overall classifiers. If we have R classifier $C_1,C_2\ldots..C_R$and their output scores are S={s_1,s_2,….S_R} for instance *x* then the MAX function will be $f_{max}(x)$:

$$f_{max}(x) = arg\ \max(S) \qquad (1)$$

This method can prove to be inefficient when skewed data sets are used. If the data is adversely either positive or negative dependent, the combiner can also become correspondingly dependent. This combiner can also become very dependent on confident classifiers and can totally isolate other classifiers. The 'MAX' functions works best with measurement level classifier outputs, but can be affected negatively by a single classifier which is performing badly.

2) Product (PROD):This method can be defined as:

$$f_{PROD}(x) = arg\ max_{i=1}^{N} \prod\nolimits_{j=1}^{K} S_{ij}\,(x) \qquad (2)$$

where: *N* is the number of classes, *x* is the input pattern, *K* represents the number of classifiers and *Sij(x)* represents the output of the i^{th} classifier for the j^{th} class for the input *x*. The advantage of this method is providing more averaged result and is less susceptible to skewed data sets. Whereas the disadvantageis its requirement to assume that the classifiers are conditionally statistically independent and susceptible to poorly performing classifiers affecting overall performance.

3) Majority voting (MAJ):In this method each classifier gives its predicted class tag for an instance *x*, the winner class is the most predicted class tag. This method is particularly successful when the classifiers provide binary output binary votes and the number of votersare odd, but in multi-class problems the simple majority may be not useful, because it can face some of confusion when making the final decision.Also MAJ method has bad performance if some classifiers are very good or very bad that why they proposed the weighted majority vote.

4) Weighted majority vote (W-MAJ):In this method the weights $w_{\,i}, i = 1, \ldots, K$ can be derived by minimizing the error of the different classifiers on the training set. Some weighted vote approaches use the overall F-measure or accuracy of a classifier on the dataset as its vote. Classifiers that perform well globally thus have a bigger influence in every vote. In another way the classifier vote for one particular class is weighted by its F-measure on that particular class. The weight of a classifier thus depends on its performance for the class it is voting for. In our experiment we used the overall and the individual precision, recall and f-measure as weighed value. In the NER task the output of classifiers are string label or class tag such as LOC, ORG, MISC and PERS, this string output and multi-class problem added difficulties to the voting method; to solve this issue in our system we used classification labels that are represented as c-dimensional binary vectors $[c_{i,1}, \ldots\ldots, c_{i,m}]^T \in \{0,1\}^m$, $i = 1, \ldots\ldots, L$ where L is the number of classifiers (C),m the number of classes, $c_{i,j} = 1$ if C_i labels x in class j (ω_j) and 0 otherwise. The binary majority vote fusion function is given by:

$$f_{maj}(x) = \arg max_{j=1}^{m} \sum_{i=1}^{L} c_{i,j} \qquad (3)$$

With this binary voting we can use the weighting MAJ in easy way by simply multiplying the weighted value by the binary vector.

5) Average voting (AVG):Each classifier output is represented as an array which contains numbers between 0 and 1 representing its confidence on the compatibility of the given input pattern with each output class (i.e. at a measurement level). The fusion function adds the votes for each output class and selects the class with the highest vote as the winner and is given by:

$$f_{AVG}(x) = arg\ max_{i=1}^{N}(\frac{1}{K}\sum_{j=1}^{K} S_{ij}(x)) \qquad (4)$$

6) Weighted average vote (W-AVG): The weighted average method is similar to the average vote method. The only difference is that each classifier is assigned a weight which is associated with its output. The weights usually represent an extra confidence or significance that the combiner assigns to each classifierand is given by:

$$f_{W-AVG}(x) = arg\ max_{i=1}^{N}(\frac{1}{K}\sum_{j=1}^{K} w_i * S_{ij}(x)) \qquad (5)$$

The average method has an advantage that it can be used as confidence based or non-confidence based and consequently, over trained and under trained classifiers can be adequately weighted to nullify extra sensitiveness. The drawback is its sensitivity towards skewed classifier values of voting[27].

7) Maximum voting (MV):This method also works well with classifiers outputting real values. Here, the most confident classifier is trustedand is given by:

$$f_{MV}(x) = arg\ max_{i=1}^{N}(\ S_{ij}(x)) \qquad (6)$$

The drawbacks of this method are its trust in the most confident classifier, which is bad if some classifiers are badly trained and its sensitivity to over-confident base classifiers.

8) Bayesian method: This method considers a probabilistic method. It assumes that the classifiers are mutually independent given a class label (conditional independence). For data set *Z* with cardinality *N*, each classifier *Ci*, a *c x c* confusion matrix *CMi* is calculated by applying *Ci* to the training data set. The *(k, s)*th entry of this matrix, $M_{k,s}^{i}$ is the number of elements of the data set whose true class label was ω_k, and were assigned by *Ci* to class ω_s. By *Ns* we denote the total number of elements of *Z* from class ω_s. Taking $M_{k,s}^{i}\ /\ N_k$ as an estimate of the probability $P(s_i|\omega_k)$, and $\frac{N_k}{N}$as an estimate of the prior probability for class ω_k,so we can use equation (7) to calculate the classifiers predictions as:

$$\mu_k(x) \propto 1/N_k^{L-1}(\prod_{i=1}^{L} CM_{ks_i}^{i}) \qquad (7)$$

where$L$ is the number of classifiers, N is the number of training data points, $k$is the number of classes,s_i is the decision of classifier i, $\mu_k$is the membership rule for label x in ω_k.This approach suffers from three limitations[28]. Firstly, it is only valid if all the classifiers can capture mutually exclusive and exhaustive possibilities on how the data was generated. Secondly,the calculation of the marginal likelihood is usually difficult.Finally, the fact that all classifiers do not usually start with the same prior assumptions. To rectify these limitations, [28]proposed a Bayesian combiner which does not assume any of the classifiers to be the best and do not expect the classifiers to be probabilistic.

All the previously presented classifiers fusion methods are static combining approaches, in the sense that the combiner decision rule is independent of the feature vector. Static approaches can be broadly divided into non-trainable and trainable. Another combining strategy is adaptive combining where the combiner is a function that depends on the input feature vector. Thus, the ensemble implements a function that is local to each region in feature space. This divide-and-conquer approach leads to modular ensembles where relatively simple classifiers specialize in different parts of the input-output space. Note that, in contrast with static-combiner ensembles, the individual model here do not need to perform well for all inputs, only in their region of expertise.

9) Bagging and Boosting:Word bagging is an acronym for Bootstrap aggregating. The idea of bagging is that the classifiers in the ensemble are trained using different training sets. In practice, only one training set can be collected, therefore, a process of selecting random subsets of the training set should be performed; this guarantees the diversity of the classifier ensemble. To make a better use of the variability in training data the base classifier has to be an unstable classifier, such that a small variation in the training data would lead to large variation in classifier output. For example NN and DT are unstable, while KNN is considered a stable classifier [29].
Boosting on the other hand answers the question of "Can a set of weak classifiers create a single strong classifier?" Boosting algorithm starts with assuming equal error distribution for all classifiers in the ensemble, because there is no prior indication about the performance of each classifier. The algorithm proceeds iteratively creating weak classifiers and updating classifier distributions according to classification accuracy. Each iteration the algorithm assigns more weight to the examples that were misclassified in previous iteration and uses these weights to create new classifier. The final decision is made using a weighted majority voting rule for all the classifiers created during the algorithm operation [30].

10) Random Forest:Random forests are a combination of tree predictors such that each tree depends on the values of a random feature vector sampled independently and with the same distribution for all trees in the forest. After a number of trees are generated, they vote for the most popular class. The idea of random forest was introduced by Leo Breiman [31]. It considers a refinement of bagging where at each tree split, a random sample of m features is drawn, and only those m features are considered for splitting. Typically $m = log_2 p$, where p is the number of features. For each tree grown on a bootstrap sample, the error rate for observations left out of the bootstrap sample is monitored. This is called the “out-of-bag” error rate. A random forest tries to improve on bagging by “de-correlating” the trees. Each tree has the same expectation [31]. Compared with Boosting there are some pros and cons.The *pros*: it is more robust, faster to train (no reweighting, each split is on a small subset of data and feature), can handle missing/partial data and is easier to extend to online version, whereas the *cons*: the feature selection process is not explicit, feature fusion is also less obvious, on small size training data and has weaker performance data.

11) Stacking combining Architecture:Stacked generalization or stacking is a layered architecture framework[32]. Stacking is a technique for achieving the highest generalization accuracy. In stacked combination architecture the classifiers at the first layer receive as input the original data, and each classifier outputs a prediction for its own sub problem. Successive layers receive as an input the predictions of the layer immediately preceding it. A single classifier at the top level outputs the final prediction. In our approach, Arabic NER task we used three classifiers in first layer which are CRF,YamCha and SVM^{hmm} and two classifiers in final layer which are DT, BN and we used Weka stacking implementation to apply three layer stacking. The stacking combination was used with named entity in different languages such as English, German, Japanese and Chinese and results show the stacking can be especially beneficial to theNER task [12].

D. Classifiers Selection

We used five folds cross-validation to choose the smallest number of classifiers which achieved a good accuracy with acceptable running time. This process has resulted in choosing five out of nine classifiers according to some measurements. These measurements were as follows: the running time, the F-measure and the Sensitivity of each model. The final output resulted in choosing the models listed in table 10 with the labels Selected1 and Selected2.

4 EVALUATION DATA

ANERcorp: Arabic Named Entity Recognition corpus[4] is a corpus of 150277k words manually annotated for the NER task developed by Benajiba following the CoNLL 2003[33]task definition. The same classes that were defined in the MUC-6 ORG, LOC and PERS were used in this corpora. MISC. is the single class that was added for NE's which do not belong to any of the other classes. This corpus is freely available and has been used for many recent research of

ANER.The tagging system that we use in our experiment follow tagged according to the IBO2 annotation; where each token of corpus is tagged as belonging to one of the following nine classes.(B-PERS,I-PERS,B-ORG,I-ORG,B-LOC,I-LOC,B-MISC,I-MISC) and (O)witch refer to the word is not a NE (Other).

/*ALTEC-NE : Recently with an objective to boost the ANER research efforts the Arabic Language Technology Center (ALTEC) has developed the ALTEC-NE a large name entity corpus that contain five million words/token manually tagged to 103 tag set organized in three level hierarchy, more details in[34]. This corpus is completely compiled from the Arabic Wikipedia and classified according to Dewy Decimal classification system. Only 280k words are freely available on ALTEC website. The tag of the annotated term begins and ends in "ne" (standing for Named Entity) and every letter after "ne" represents a level in the hierarchy structure according to the design of the tag set. For example “Alexandria” is tagged as <nelgc>*Alexandria*</nelgc> to denote that it is a named entity, Location, Geographical, and City. ALTEC NE's tags organized in a hierarchy from three levels: level one contains 15 classes or types: person, organization, location, facility, product, disease, event, title, job, god, nationality, natural object, color, numex and timex.

5 EXPERIMENTS AND RESULTS

We evaluated our ANER system using the two ALTEC-NE and ANERcorp. The system performance is evaluated in terms of the standard recall, precision and F-measure parameters as defined below:

$$Recall = \frac{\text{NE's retrieved by the system}}{NE's\text{ present in the test set}}$$

$$Precision = \frac{\text{NEs correctly retrieved by the system}}{\text{NEs retrieved by the system}}$$

$$F - measure = 2 * \frac{Recall * Precision}{Recall + Precision}$$

In order to achieve clean corpus some pre-processing operations are applied on the original corpus.We used some orthographic normalizations as follow: Change all alif forms to bare alif, Map ya to alif maqsura, Strip all diacritics according to our needs and Strip tatweel elongation character. ALTEC-NE corpus is in the form of raw text files with stand-off annotation in XML files. We implemented an XML parser to extract text and annotation from the standoff annotated corpus and convert it to IBO2 format to meet the ANERcorp annotations for comparison reasons and to meet the requirement of classifier toolkits data input. ALTEC-NE areorganized in three level hierarchy and we chose to work on level one with only five NE classes.

To determine the effective feature set for the ANER system we conducted thirty one different experiments. For this set of experiments we used the CRF classifier since it requires less time for training and testing compared with SVM. For unbiased results we evaluate the performance in our experiments using 6-fold cross validation. Table I describes the different combinations tested with the corresponding average F-measure of the developed ANER system.

For the comparison of our ANER system with state of art systems we found the best reported result for ANER system using CRF was (F_B= 79.2%)[6]on the ANERCorp using the set of lexical, CXT, GAZZ, POS-tag, BPC and NAT features. For SVM based systems we found the best reported result is (F_B=80.4%)[7]. Using the set of CXT, lexical, morphological, Gazz, POS-tags and BPC, NAT and the corresponding English capitalization features and the best result obtained when using SVM^{hmm}is F_B=83.2%[10] and we considered that result as our baseline.

Since all comparisons were made with all the models trained and tested on same datasets, the relative performance should largely be reflective of reality. However, the rates attained for the models, on a general scale, might be slightly different than expected. This could be due to several factors most notably of which is the errors introduced to the data during the preprocessing phase; the segmentation and POS tagging phases applied to data added further noise to the data used due to the propagation of errors between phases of the pipeline.Also we have noticed that some of features are more effective for one corpus than in other such as $PW_{(GAR,NEDA)}),CD,WL$.

In the first experiment we kept the punctuations in the data, because it is sometimes used as an indicator for the presence of NE’s such as in ANERcorp the quotation symbol (“) comes before locations NE’s 30 times and 50 times comes before person NE’s also it comes 99 times before organizations. Also the comma (،) comes 61 times before location NE’s and 53 before person NE’s. Similarly in ALTEC-NE corpus the (،) comes 890, 315, 94, 651 times before person, location, organization and Misc NE’s types, respectively. In contrast the comma(،) comes 9,889 before non NE’s in ALTEC-NE corpuswhichraise the ambiguity,so the large presence of punctuations and symbols in corpus may add noise to the data which affect the extraction process of the NE features.

TABLE I
CORRESPONDING F-MEASURE FOR DIFFERENT SET COMBINATIONS OF FEATURES APPLIED ON TWO CORPUS ANERCORP AND ALTEC USING CRF CLASSIFIER

	Features set combinations	**F-Measure on CRF result**	
		ALTEC-NE	**ANERcorp**
1	CXT	72.3	64.52
2	CXT,POSStanford	73.16	66.64
3	CXT, POSAMIRA-2.1	72.78	67.06
4	CXT, POSstanford, POSAMIRA-2.1	74.65	69.4
5	CXT, BPC	72.85	66.42
6	CXT, POSstanford,DT	73.31	66.87
7	CXT, POSstanford,DT,CD	73.34	66.82
8	CXT, POSstanford,DT,WL	73.28	66.97
9	CXT, POSstanford,DT,CD,WL	73.82	66.98
10	CXT, POSstanford,DT,SW	73.26	67
11	CXT, POSstanford,DT,CD,WL,SW	73.92	67.77
12	CXT, POSstanford,DT,CD,WL,SW,POSS	74.23	67.94
13	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ	82.03	81.69
14	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT	82.17	81.95
15	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT,PW(GAR,NEDA)	82.01	81.88
16	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA), NOT	82.44	82.06
17	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA),NOT, MFI	83.20	82.18
18	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA),NOT, MFI,CH	84.11	84.65
19	CXT, POSstanford,DT,GAZ,NAT, NOT,MFI, CH,CW	83.99	84.86
20	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA), NOT,MFI,CH,PW,NW	83.67	84.19
21	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA), NOT,MFI,CH,PW,NW, PWCXT+1/-1	83.53	84.53
22	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR, NEDA), NOT,MFI,CH,PW,NW,PWCXT+1/-1,CW	83.65	84.59
23	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA), NOT, MFI,CH,CW	84	84.47
	2^{3}+BPC,POSAMIRA	83.72	84.41
24	CXT, POSstanford, POSAMIRA-2.1, DT,CD,WL, SW, POSS, GAZ,NAT, PW(GAR,NEDA), NOT, MFI,CH	83.65	83.41
25	CXT,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA),NOT, MFI,CH, POSAMIRA-2.1	83.04	83.61
26	CXT, POSstanford,DT,GAZ,NAT, NOT,MFI,CH,POSAMIRA2.1	83.69	83.43
27	(CXT, POSstanford, POSAMIRA-2.1,DT,CD,WL, SW,POSS,GAZ, NAT, BPC,PW(GAR,NEDA), NOT,MFI,CH,PW, NW) without MADA	83.66	84.93
28	(CXT, POSstanford, POSAMIRA-2.1,DT,CD,WL, SW,POSS,GAZ, NAT, BPC,PW(GAR,NEDA), NOT,MFI,CH,PW, NW) with MADA	84.21	85.08
29	CXT, POSstanford,DT,CD,WL,SW,POSS,GAZ,NAT, PW(GAR,NEDA),NOT, MFI,CH,MADA	84.44	85. 20
30	CXT, POSstanford, DT,CD,WL, SW,POSS,GAZ, NAT, PW(GAR,NEDA), NOT,MFI,CH,PW,NW,MADA	84.21	84.9
31	(CXT, POSstanford, DT,CD,WL, SW,POSS,GAZ, NAT, PW(GAR,NEDA), NOT,MFI, CH,PW,NW,MADA) with use AMIRA tokenization	**84.68**	**85.25**

The PW,NW and CW features were expected to provide positive effects but results on Table Ishow that these features have negative impact. This may be due to the noise that still exists in the data or the effect of the tokenization errors. We investigated several tokenization toolswhich are AMIRA2.1 and MADA+TOKEN.3.2. Rows 27 to 31TableIshow the impact of these tools. If the affixes can be identified accurately then the impact of PW, NW and DT features would be better. In the figure below the word “اليم” comes in two different meanings once as a noun “the sea” and other one as an adjective “painful”. So when using inaccurate tokenization the feature Determent DT become not useful but it gets much better positive effect with the more accurate tokenization. The same effect for prepositions and appeal letters like the word “وبمحمد”. Also sometimes non-Arabic names such as the translated names may be similar to Arabic words for example:(“ للمتحف لن هوى شيانج تبرع بها عالم الآثار الصيني”) the word "لن"come here as Chinese person name but in Arabic language it has the meaning "not allows"which is member of the "not preceding Arabic NE’s" list that prevent the follower word to be NE as in this case it has to be followed by a verb. Also we found the performance ofStanford POS andAMIRA2.1 POS taggingarevery close but Stanford is faster in tagging time. From results in TableI we can see that the best performing feature set is row 31 and this result was consistent for both used databases ALTEC-NE and**ANERcorp**.

All base classifiers are trained on the features listed in row 31 from TableI and evaluated using 5-fold cross validations; the best individual performing classifier in terms of F-measure (F) and accuracy (acc) was the decision tree with F=87.29, acc=98.38 on ANERcorp and F=87.22, acc=96.54 on ALTEC-NE corpus. Classifiers performances are presented in TableII.

TABLE II
OVERALL F-MEASURE, ACCURACY AND RUNTIME IN SECONDS FOR EACH INDIVIDUAL BASE CLASSIFIER OBTAINED ON ALTEC-NE AND ANERCORP CORPUS USING 5-FOLD.

Model	**ANERcorp**		**ALTEC-NE**		**Time for processing the 30Kwords test samples (seconds)**
	F	**Acc.**	**F**	**Acc.**	
SVM^{hmm}	83.42	96.9	81.06	93.6	2.3
YamCha	85.24	97.5	82.5	94.1	228
CRF	84.67	97.4	86.61	95.5	0.83
DT	87.29	98.4	87.22	96.5	0.36
KNN(K=3	82.83	97.6	83.2	95.5	3191.88
SMO	84.22	98.1	78.48	94.4	4.64
Logistic	83.71	97.95	78.23	94.4	1.41
Naïve B	75.07	96.2	57	84	4.80
Bayes net	75.12	95.4	73.37	90.5	1.59

The results in Table II motivated us to think about classifiers fusion to benefit from the diversity in the classifiers output. Table IIIshows the results of using the voting fusion method between all the nine classifiers.We can see that the voting combination method has led to improvement in system performance by 3.21 % in term of F-measure on ANERcorp and by 2.54% on ALTEC-NE. Also results in Table III show that using weighted voting provide some minor improvement with the "overall recall" as the best weighing scheme.

TABLE III
MAJORITY VOTING RESULTS FOR ALL NINE BASE CLASSIFIERS (DT,YAMCHA,CRF,BN, NB, SMO, LOGISTIC, KNN, SVM^{HMM}) IN TERM OF F-MEASURE AND ACCURACY (ACC.)

Majority vote (MAJ) and Weighted MAJ (W-MAJ)	**ANERcorp**		**ALTEC-NE**	
	F	**Acc.**	**F**	**Acc.**
best individual classifier	87.29	98.38	87.22	96.54
Normal MAJ	90.25	98.62	89.53	97
Binary MAJ	90.32	98.64	89.17	96.97
W-MAJ by over all accuracy	90.38	98.64	89.64	97.07
W-MAJ by overall F-measure	90.45	98.64	89.65	97.07
W-MAJ by overall precision	90.4	98.63	89.62	97.04
W-MAJ by overall recall	**90.49**	**98.66**	**89.75**	97.11
W-MAJ by F-measure for each class	89.77	98.5	89.61	97.05
W-MAJ by precision for each class	89.98	98.53	90.12	**97.15**
W-MAJ by recall for each class	89.77	98.51	89.41	97

Table IVshows the results for applying the Average (AVG) and weighted AVG fusion methods on the nine classifiers. The results show that the weighted AVG by overall precision gave the best result which has an improvement in system F-measure by 4.02% on ANERcorp and 4.55% on ALTEC-NE corpus. When we use term weighted average by F-measure, precision or recall for each class in table IV we mean for each NE class we calculate those measures and use them as weighted value. The weighting process is performed via multiplying the class measure value like precision by the classifier output probability for the same class.

TABLE IV
AVERAGE AND WEIGHTED AVERAGE FUSION METHODS RESULTS FOR 9 CLASSIFIERSIN TERM OF F-MEASURE AND ACCURACY

Average vote (AVG) and Weighted AVG (W-AVG)	**ANERcorp**		**ALTEC-NE**	
	F	**Acc.**	**F**	**Acc.**
best individual classifier	87.29	98.38	87.22	96.54
Average vote (AVG)	91.05	98.63	90.93	97.35
W-AVG by over all accuracy	91.08	98.63	91.12	97.42
W-AVG by overall F-measure	91.28	98.65	91.52	97.51
W-AVG by overall precision	**91.31**	98.65	**91.75**	97.58
W-AVG by overall recall	91.27	**98.67**	91.19	97.48
W-AVG by F-measure for each class	90.9	98.54	91.07	**97.59**
W-AVG by precision for each class	91	98.56	91.43	97.47
W-AVG by recall for each class	90.81	98.52	90.13	97.18

Table V shows the results for using Max, Min and the nonlinear fusion methods like product, max vote, and Bayesian.From the results in table V we can see that we have a slightly positive impact compared to the previous methods on ANERcorp. Also those methods have high sensitivity to bad classifiers and the scoring value for each class.

TABLE V
RESULTS of PRODUCT, MAX. VOTE,MAX,MIN and BAYESIAN FUSION METHOD USING NINE CLASSIFIERS

Non-linear methods	**ANERcorp**		**ALTEC-NE**	
	F	**Acc.**	**F**	**Acc.**
Best individual classifier	87.29	98.38	87.22	96.54
Product	90.15	98.6	88.62	96.71
Max Vote	88.86	98.41	88.43	96.53
Bayesian	88.9	98.52	88.74	96.63
Maximum	87.8	98.11	87.39	95.74
Minimum	84.22	98.04	78.48	94.43
Maximum using acc. As confidence	87.29	98.38	87.22	96.54
Maximum using overall F. As confidence	90.1	98.27	88.30	95.72
Max. using F. for each class as confidence	80.66	96.77	74.81	92.39
Max. using overall precision as confidence	87.96	98.27	87.30	95.72

Table VIshows the results obtained when using the two layers and three layers Stacking fusion method. Results in table VI shows that the stacking method has a significant improvement in ANER system accuracy. It outperforms the base classifier by 5% in overall F-measure forANERcorp and 4.23 % for ALTEC-NE for level two. When we move to level three the performance slightly increases.

TABLEVI
STACKING RESULTS

Stacking	**ANERcorp**		**ALTEC-NE**	
	F	**Acc.**	**F**	**Acc.**
Best on 1st layer	85.24	97.48	86.61	95.47
Layer 2 (NB)	88.60	98.04	86.73	95.54
Layer2 (DT)	90.78	98.39	88.93	96.26
Layer 3(KNN)	90.89	98.46	89.39	96.52

Table VIIpresents the results of Bagging,Random forest and Adaboost.M2 ensemble methods using decision tree as base classifier. We can see the obtained result is better than best individual DT classifier. Figure 1 displays the boosting error on different number of trees.

TABLE VII
BAGGING, RANDOM FOREST and BOOSTING RESULTS

Bagging, Random Forest and Boostingwith DT	**ANERcorp**		**ALTEC-NE**	
	F	**Acc.**	**F**	**Acc.**
best individual classifier	87.29	98.38	87.22	96.54
10 itter. Bagging	88.79	98.60	88.63	96.92
10 trees Random Forest	88.46	98.76	88.76	96.93
17 iteration (Adaboost.M2)	89.74	98.74	88.7	96.91

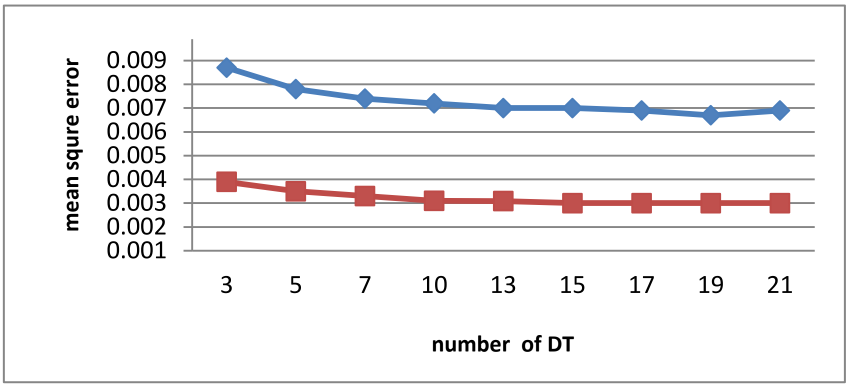

Figure 1:Adaboost.M2 train and estimate the generalization error

Finally we run set of experiments to select the best group of classifiers to be fused for the ANER system. Table VIIIshows that using only five models out of the nine experimented models provide the best performance. The cross-validation has excluded the weak models which negatively affect the overall results and this explains the enhancement of the achieved results.

To summarize the whole set of experiments figure 2 compares between the different methods of classifiers fusion depending in the F-measure. As shown figure 2 we notice that all the fusion methods have achieved a better result than the best individual classifier except the Min fusion method. The W-AVG fusion method has achieved the highest result and Min fusion method has achieved the least F-measure of all other methods.

TABLE VIII
CLASSIFIERS SELECTION RESULTS

Models	**F-measure**			
	AVG		**MAJ**	
	ANERcorp	**ALTEC**	**ANERcorp**	**ALTEC**
Selected: DT, CRF, Logstic, SVM^{hmm} , SMO	92.11	92.07	90.38	89.94
Selected2: DT, CRF, SMO	88.77	90.64	89.55	90.11
Best three in Time : DT, CRF, LOG	90.07	90.89	89.58	90.15
Best five in Time : DT, CRF, LOG, BN, SVM^{hmm}	91.9	90.9	90.49	90.26
Best three in F: DT, YamCha, CRF	88.21	90.32	88.14	89.47
Best 5 in F.: DT, YamCha, CRF, SMO, Logistic	90.12	91.6	90.00	90.16
Best three in sensitivity: DT, SVM^{hmm}, SMO	91.33	87.11	89.2	89.2
Best 5 in sensitivity: DT, SVM^{hmm}, SMO,BN, logistic	91.73	88.89	89.51	86.05

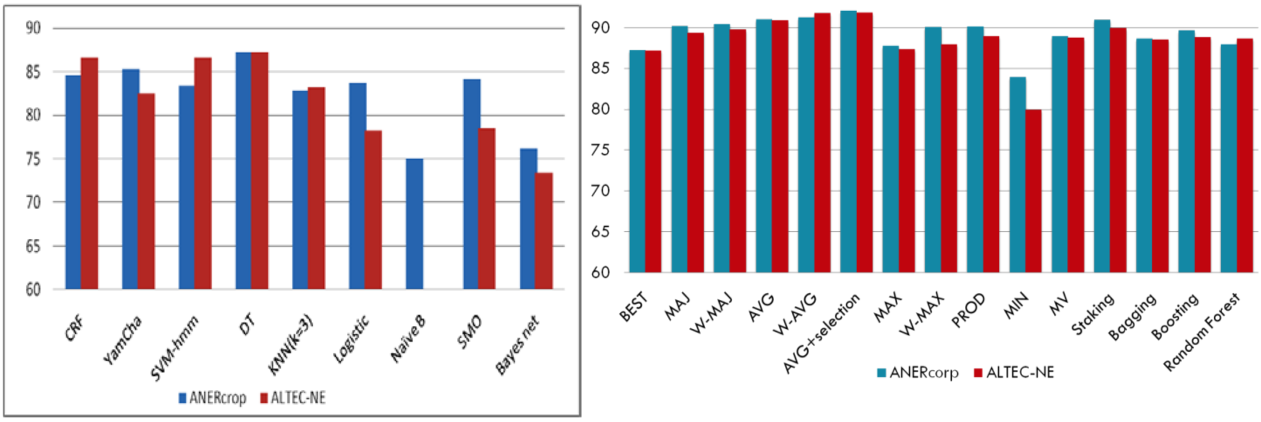

Figure 2: Individual and Fusion Methods Comparison Results Using F-Measure

6 CONCLUSIONS

In order to solve the problem of ANER a new system was presented using the fusion between nine classifiers trained on two Arabic corpus and different sets of features that are both language independent and language specific. About ten fusion methods have been experimented individually. This system involve four phases as follow: feature extraction and selection, designing the base classifier, fusion methods and classifiers selection. Our experiments show that all fusion methods outperform the best individual classifier except the Min fusion method. The W-AVG fusion method has achieved the highest result after we apply classifier selection, which were F=92.11% using fusion between the five classifiers of DT, CRF, Logistic, SVM_hmm and WEKA SVM model (SMO). In future we are planning to use a larger Arabic corpuses annotated for ANER and to extend our system to deal with unstructured and Colloquial Arabic that is most commonly found in social media such as Twitter.Also we plan to investigate some semi-supervised training approaches.

REFERENCES

[1] K. Shaalan and H. Raza, "Arabic Named Entity Recognition from Diverse Text Types," *Advances in Natural Language Processing*, pp. 440-451, 2008.

[2] S. AbdelRahman, M. Elarnaoty, M. Magdy, and A. Fahmy, "Integrated Machine Learning Techniques for Arabic Named Entity Recognition," *International Journal of Computer Science Issues (IJCSI),* vol. 7, no. 4, pp. 27-36, July 2010.

[3] Y. Benajiba, M. Diab, and P. and Rosso, "Arabic Named Entity Recognition: A Feature-Driven Study," *Audio, Speech, and*

Language Processing, IEEE Transactions on, vol. 17, no. 5, pp. 926 - 934, July 2009.

[4] Y. Benajiba, P. Rosso, and i. J. M. Ruiz, "ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy," *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing-2007)*, pp. 143-153, 2007.

[5] Y. Benajiba and P. Rosso, "ANERsys 2.0: Conquering the NER task for the Arabic language by combining the Maximum Entropy with POS-tag information," *Proceedings of Workshop on Natural Language-Independent Engineering, 3rd Indian International Conference on Artificial Intelligence (IICAI-2007)*, pp. 1814-1823, 2007.

[6] Y. Benajiba and P. Rosso, "Arabic Named Entity Recognition Using Conditional Random Fields," *CICLing '07 Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, 2008.

[7] Y. Benajiba, M. Diab, and P. Rosso, "Arabic Named Entity Recognition: An SVMBased Approach," *Proceedings of Arab International Conference on Information Technology (ACIT 2008)*, 2008.pp 154.

[8] Y. Benajiba, M. Diab, and P. Rosso, "Using Language Independent and LanguageSpecific Features to Enhance Arabic Named Entity Recognition," *The International Arab Journal of Information Technology*, vol. 6, no. 5, pp. 464- 473, 2009.

[9] A. Abdul-Hamid and K. Darwish, "Simplified Feature Set for Arabic Named Entity Recognition," *Proceedings of the 2010 Named Entities Workshop, (ACL 2010)*, pp. 110-115, 2010.

[10] R. Koulali and A. Meziane, "A Contribution to Arabic Named Entity Recognition," *Tenth International Conference on ICT and Knowledge Engineering,IEEE*, pp. 46-51, 2012.

[11] N. F. Mohammed and N. Omar, "Arabic Named Entity Recognition Using Artificial Neural Network," *Journal of Computer Science 8 (8): 1285-1293, 2012*, 2012.

[12] R. Florian, A. Ittycheriah, H. Jing, and T Zhang, "Named Entity Recognition through Classifier Combination," *In CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, vol. 4, pp. 168-171 , 2003.

[13] H. Wang, T. Zhao, H. Tan, and S. Zhang, "Biomedical Named Entity Recognition Based on Classifiers Ensemble," *International Journal of Computer Science & Applications*, vol. 5, no. 2, pp. 1-11, 2008.

[14] A. Danesh, B. Moshiri, and O. Fatemi, "Improve text classification accuracy based on classifier fusion methods," *Information Fusion, 2007 10th International Conference on 9-12 July*, pp. 1-6, July 2007.

[15] C. Wei Wu, S. Jan, T. Tsai, and W. Hsu, "On Using Ensemble Methods for Chinese Named Entity Recognition," *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pp. 142--145, July 2006.

[16] A. and Saha,S. Ekbal, "Maximum entropy classifier ensembling using genetic algorithm for NER in Bengali," *Proceedings of the International Conference on Language Resources and Evaluation (LREC).*, may 2010pp 512..

[17] B. Desmet and V. Hoste, "Dutch named entity recognition using classifier ensembles," *In BNAIC(2011) Proceedings of the 23rd Benelux conference on artificial intelligence , Belgian/Netherlands Artificial Intelligence Conference*, pp. 387–388, November 2011.

[18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support Vector Machine Learning for Interdependent and Structured Output Spaces," *In Proceedings of the 21 International Conference on Machine Learning (ICML '04), 104. Banff, Alberta, Canada*, 2004. pp 196..

[19] A. Ekbal and S. Bandyopadhyay, "Named Entity Recognition using Support Vector Machine: A Language Independent Approach," *International Journal of Electrical and Electronics Engineering*, vol. 4, no. 2, pp. 155-170, May 2010.

[20] Mon Diab T., "Second Generation Tools (AMIRA 2.0):Fast and Robust Tokenization,POS tagging, and Base Phrase Chunking," *Proceedings of the Second International Conference on Arabic Language Resources and Tools.The MEDAR Consortium*, April 2009.

[21] N. Habash and O. Rambow, "Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop.," *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 573–580, June 2005.

[22] Y. Altun, I. Tsochantaridis, and T. Hofmann, "Hidden Markov Support Vector Machines," *In Proceedings of the Twentieth International Conference on Machine Learning (ICML '03)*, vol. 20, no. 3, 2003.

[23] Mai M. Oudah and K. Shaalan, "A Pipeline Arabic Named Entity Recognition Using a Hybrid Approach," *COLING*, 2012.

[24] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables," *Journal of Machine Learning Research (JMLR)*, pp. 1453-1484, 2005.

[25] T. Joachims, T. Finley, and C. Yu, "Cutting-plane Training of Structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[26] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of combining multiple classifiers and their application to handwriting recognition," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 3, pp. 418–435, 1992.

[27] R. Ranawana and V. Palade, "Multi-Classifier Systems: Review and a roadmap for developers," I*nternational Journal of Hybrid Intelligent Systems*, vol. 3, no. 1, pp. 35 - 61, January 2006.

[28] Z. Ghahramani and H. Kim., "Bayesian classifier combination," *Gatsby Technical Report*, September 2003.

[29] L. Breiman, "Bagging Predictors," *Univesity of California, Dept. of Statistics, Technical Report No. 421*, September 1994.

[30] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and Computation*, vol. 121, no. 2, pp. 256–285.

[31] L. Breiman, "RANDOM FORESTS," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[32] D. H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–260, 1992.

[33] F. D. Meulder, "CONLL-2003: List of tags with associated categories of names," *CoNLL*, 2003.

[34] S. Alansary, "ALTEC Named Entity Corpus for Modern Standard Arabic," *icca*, 2012.

BIOGRAPHY

Wasim M. Abdulwasea received his B.Sc. in 2010 from University of Yemen and a M. Sc from faculty of Computers and Information at Cairo University in 2014. His research interest is in Arabic Language Processing and Text analysis.

Sherif Mahdy Abdou received his B.Sc. and M.Sc. degrees in computer science and automatic control from University of Alexandria, Egypt in 1993 and 1997, respectively. He received a Ph.D degree in Electrical and Computer Engineering from University of Miami, USA in 2003. In 2003 Dr. Abdou joined BBN Technologies as a senior staff scientist in the Arabic language team of the Ears project to provide affordable reusable speech-to-text decoding for the Defense Advanced Research Projects Agency, DARPA. In 2005 Dr. Abdou was appointed as the research and development manger of the Research and Development Internatinal (RDI) company where he is leading a team to develop several products for natural language processing, computer aided language learning, speech recognition, speech syntheses, optical character recognition, handwriting recognition with special focus on the technologies of the Arabic language.In 2005 Dr. Abdou joined Cairo University as an Assistant Professor at the Information Technology department at the Faculty of Computers and Information. Dr. Abdouis one of the holders of the patent “ Systems and Methods for Quran Recitations Rules: HAFSS”. Dr. Abdou is a member of the review committee in several conferences andjournals in the HLT fieldsand is the Principal Investigator and Co-Principal Investigator of several research projects in the areas of Language learning, Virtual tutors, Web monitoring and Intelligent Contact Centers.

Hassanin M. Al-Barhamtoshy received the B.S. degree in electronic and communication engineering from Cairo University, in 1978, and the M.S. degree in systems and computers engineering from the Al-Azhar University, Cairo, 1985. In 1992, he received the Ph.D. degree in systems and computers engineering from Al-Azhar University, Cairo. During 1992–1997, he was an Assistant Professor in the Department of Systems and Computer Engineering at Al-Azhar University. During 1996-1997 he was an Assistant Professor in Computer Science at KAU University, Jeddah, Saudi Arabia. During 1998-2002 he was an Associate Professor in Computer Science at KAU University, Jeddah, Saudi Arabia. He is currently Professor in the Department of Computer Science and Information Technology at Faculty of Computing and Information Technology, KAU University(2003-now). His research interests include language processing and machine translation, image processing, software engineering, intelligent systems, speech processing, e-learning, and RFID.

التعرف على أسماء الأعلام في النصوص العربية باستخدام الدمج بين اكثر من مصنف آلي

[*]حسنين البرهمتوشى[**3]شريف مهدى عبده , [*2]وسيم عبد الواسع , [1]

كلية حاسبات ومعلومات جامعة القاهرة. جمهورية مصر العربية[*]

[1]w.abdulwasea@gmail.com
[2]s.abdou@fci-cu.edu.eg

كلية علوم حاسب جامعة الملك عبد العزيز , جدة, المملكة العربية السعودية[**]

[3]hassanin@kau.edu.sa

الملخص :

مؤخرا أصبح استخراج وتصنيف أسماء الأعلام في النص (على سبيل المثال الأشخاص الاماكن ، والمنظمات) تعتبر ذاتأهمية كبرى في معالجة اللغات الطبيعية, في هذة الرسالة تم تقديم طريقة جديدة من أجل تحسين طريقة التعرف على أسماء الأعلام في اللغة العربية, هذه الطريقة تستخدم عناصر تعتمد على اللغة العربية و عناصر أخرى لا تعتمد عليها , تم توظيف هذه العناصر في نظام تعلُّم آلي تمييزي يعتمد على استخدام أكثر من مصنف آلي. يتم دمج جميع هذه المصنِّفات بإحدى الطرق التالية: طريقة تصويت الأغلبية , طريقة أخذ المتوسط , طريقة أخذ القيمة العظمى , طريقة التعزيز , طريقة تقسيم البيانات , طريقة الأشجار العشوائية , طريقة التكديس و طريقة حاصل الضرب . تم تجربة جميع طرق الدمج السابقة كل على حده و جميعها أتت بنتائج أعلى من نتائج جميع الأبحاث المنشورة مسبقا في هذا المجال , حيث تم الحصول على أعلى النتائج عامة من خلال عملية الدمج عن طريق أخذ المتوسط . تم تقييم النظام المقدم باستخدام قاعدتي بينات لنصوص معنونة بموضع الأسماء فيها. قاعدة البيانات الأولى تم توفيرها مؤخرا عن طريق مركز دعم اللغة العربية ALTEC, و قاعدة البيانات الأخرى تعتبر معيارية في هذا المجال و تسمى ANERcrop.