Arabic Named Entity Recognition using Optimized Feature Sets

Yassine Benajiba[•] **Mona Diab**[◇] **Paolo Rosso**[•]

[•]Natural Language Engineering Lab.,
Dept. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
{ybenajiba,prosso}@dsic.upv.es
[◇]Center of Computational Learning Systems
Columbia University
mdiab@cs.columbia.edu

Abstract

The Named Entity Recognition (NER) task has been garnering significant attention in NLP as it helps improve the performance of many natural language processing applications. In this paper, we investigate the impact of using different sets of features in two discriminative machine learning frameworks, namely, Support Vector Machines and Conditional Random Fields using Arabic data. We explore lexical, contextual and morphological features on eight standardized data-sets of different genres. We measure the impact of the different features in isolation, rank them according to their impact for each named entity class and incrementally combine them in order to infer the optimal machine learning approach and feature set. Our system yields a performance of $F_{\beta=1}$-measure=83.5 on ACE 2003 Broadcast News data.

1 Introduction

Named Entity Recognition (NER) is the process by which named entities are identified and classified in an open-domain text. NER is one of the most important sub-tasks in Information Extraction. Thanks to standard evaluation test beds such as the Automatic Content Extraction (ACE)[1], the task of NER has garnered significant attention within the natural language processing (NLP) community. ACE has facilitated evaluation for different languages creating standardized test sets and evaluation metrics. NER systems are typically enabling sub-tasks within large NLP systems. The quality of the NER system has a direct impact on the quality of the overall NLP system. Evidence abound in the literature in areas such as Question Answering, Machine Translation, and Information Retrieval (Babych and Hartley, 2003; Ferrández et al., 2004; Toda and Kataoka, 2005). The most prominent NER systems approach the problem as a classification task: identifying the named entities (NE) in the text and then classifying them according to a set of designed features into one of a predefined set of classes (Bender et al., 2003). The number of classes differ depending on the data set. To our knowledge, to date, the approach is always to model the problem with a single set of features for all the classes simultaneously. This research, diverges from this view. We recognize that different classes are sensitive to differing features. Hence, in this study, we aspire to discover the optimum feature set per NE class. We approach the NER task from a multi-classification perspective. We create a classifier for each NE class independently based on an optimal feature set, then combine the different classifiers for a global NER system. For creating the different classifiers per class, we adopt two discriminative approaches: Support Vector Machines (SVM)(Vapnik, 1995), and Conditional Random Fields (CRF)(Lafferty et al., 2001). We comprehensively investigate many sets of features for each class of NEs: contextual, lexical, morphological and shallow syntactic features. We explore the feature sets in isolation first. Then, we employ the Fuzzy Borda Voting Scheme (FBVS) (García Lapresta and Martínez Panero, 2002) in order to rank the features according to their perfor-

[1] http://www.nist.gov/speech/tests/ace/2004/doc/ace04-evalplan-v7.pdf

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 284–293,
Honolulu, October 2008.

mance per class. The incremental approach to feature selection leads to an interpretable system where we have a better understanding of the resulting errors. The paper is structured as follows: Section 2 gives a general overview of the state-of-the-art NER approaches with a particular emphasis on Arabic NER; Section 3 describes relevant characteristics of the Arabic language illustrating the challenges posed to NER; in Section 4.1 we describe the Support Vector Machines and Conditional Random Fields Modeling approaches. We discuss details about our feature-set in 4.2 and describe the Fuzzy Borda Voting Scheme in Section 4.3. Section 5 describes the experiments and shows the results obtained; Withing Section 5, Section 5.1 gives details about the data-sets which we use; finally, we discuss the results and some of our insights in Section 6 and draw some conclusions in 7.

2 Related Work

To date, the most successful language independent approaches to English NER are systems that employ Maximum Entropy (ME) techniques in a supervised setting (Bender et al., 2003).

(Tran et al., 2007) show that using a Support Vector Machine (SVM) approach outperforms ($F_{\beta=1}$=87.75) using CRF ($F_{\beta=1}$=86.48) on the NER task in Vietnamese. For Arabic NER, (Benajiba et al., 2007) show that using a basic ME approach yields $F_{\beta=1}$=55.23. Then they followed up with further work in (Benajiba and Rosso, 2007), where they model the problem as a two step classification approach applying ME, separating the NE boundary detection from the NE classification. That modification showed an improvement in performance yielding an $F_{\beta=1}$=65.91. None of these studies included Arabic specific features, all the features used were language independent. In a later study, (Benajiba and Rosso, 2008) report using lexical and morphological features in a single step model using CRF which resulted in significant improvement over state of the art to date for Arabic NER, yielding $F_{\beta=1}$=79.21. However, the data that was used in these evaluation sets were not standard sets. Most recently, (Farber et al., 2004) have explored using a structured perceptron based model that employs Arabic morphological features. Their system benefits from the basic POS tag (15 tags) information and the corresponding capitalization information on the gloss corresponding to the Arabic word. Exploiting this information yields a significant improvement in recall of 7% and an overall $F_{\beta=1}$=69.6 on the ACE2005 data set. The authors note the lack of improvement in the system's performance when using other Arabic morphological information.

3 Arabic in the context of NER

The Arabic language is a language of significant interest in the NLP community mainly due to its political and economic significance, but also due to its interesting characteristics. Arabic is a Semitic language. It is known for its templatic morphology where words are made up of roots, patterns, and affixes. Clitics agglutinate to words. For instance, the surface word وبحسناتهم *wbHsnAthm*[2] 'and by their virtues[fem.]', can be split into the conjunction *w* 'and', preposition *b* 'by', the stem *HsnAt* 'virtues [fem.]', and possessive pronoun *hm* 'their'.

With respect to the NER task, Arabic poses several major challenges:

Absence of capital letters in the orthography: English like many other Latin script based languages has a specific marker in the orthography, namely capitalization of the initial letter, indicating that a word or sequence of words is a named entity. Arabic has no such special signal rendering the detection of NEs more challenging.

Absence of short vowels: The absence of short vowels renders the lexical items a lot more ambiguous than in other languages exacerbating the homography problem. The average polysemy for surface unvowelized words in Arabic is 12 possible vowelized forms and when the inflections are removed the average is 4 possible vowelized forms.[3] For instance, words such as براد *brAd* can be read both as 'refrigerator' or 'Brad',respectively, where the former is a common noun and the latter is an NE.

[2]We use the Buckwalter transliteration scheme to show romanized Arabic (Buckwalter, 2002).

[3]It is worth noting that each vowelized form could still be ambiguous as in the English homograph/homophone 'bank' case.

The Arabic language is highly inflectional: As we mentioned earlier, Arabic language uses an agglutinative strategy to form surface tokens. As seen in the example above, a surface Arabic word may be translated as a phrase in English. Consequently, the Arabic data in its raw surface form (from a statistical viewpoint) is much more sparse which decreases the efficiency of training significantly.

4 Our Approach

We approach the problem of NER from a per NE class based perspective. The intuition is that features that are discriminative for one NE class might not be for another class. In the process, we decide on an optimal set of features for each NE class. Finally we combine the different classifiers to create a global NER system. Hence, we identify a set of features for NER and proceed to investigate them individually. Then we use an automatic ranking system to pick the optimal set of features per NE class. To that end, we use the Fuzzy Borda Voting Scheme (FBVS). We employ two discriminative classification techniques: Support Vector Machines (SVM) and Conditional Random Fields (CRF). Even though some previous studies seem to point to the superiority of SVM over CRF for NER (Tran et al., 2007), it is hard to draw a definitive conclusion since their assessment was based on comparing the average F-measure.[4] Moreover, the best system to date on Arabic NER reports results using CRF (Benajiba and Rosso, 2008). We adopt an IOB2 annotation scheme for classification. For each NE class, we have two types of class labels: B-Class, marking the beginning of a Class chunk, and I-Class marking the inside of a class chunk. Finally, we mark words not participating in an NE as O, meaning they are outside some NE class label.

4.1 SVM and CRF

SVM approach is based on Neural Networks (Vapnik, 1995). The goal is to find, in the training phase, the best decision function which allows us to obtain the class c for each set of features f. SVM are robust to noise and have powerful generalization ability, especially in the presence of a large number of features. Moreover, SVM have been used successfully in many NLP areas of research in general (Diab et al., 2007), and for the NER task in particular (Tran et al., 2007). We use a sequence model *Yamcha toolkit*,[5] which is defined over SVM.

CRF are a generalization of Hidden Markov Models oriented toward segmenting and labeling sequence data (Lafferty et al., 2001). CRF are undirected graphical models. During the training phase the conditional likelihood of the classes are maximized. The training is discriminative. They have been used successfully for Arabic NER (see section 2). We have used *CRF++*[6] for our experiments.

4.2 Our Feature Sets

One of the most challenging aspects in machine learning approaches to NLP problems is deciding on the optimal feature sets. In this work, we investigate a large space of features which are characterized as follows:

Contextual (CXT): defined as a window of $+/-$ n tokens from the NE of interest

Lexical (LEX_i): defined as the lexical orthographic nature of the tokens in the text. It is a representation of the character n-grams in a token. We define the lexical features focusing on the first three and last three character n-grams in a token. Accordingly, for a token $C_1C_2C_3...C_{n-1}Cn$, then the lexical features for this token are LEX_1=C_1, LEX_2=C_1C_2, LEX_3=$C_1C_2C_3$, $LEX4$=C_n, $LEX_5 = C_{n-1}C_n$, $LEX_6 = C_{n-2}C_{n-1}C_n$.

Gazetteers (GAZ): These include hand-crafted dictionaries/gazetteers listing predefined NEs. We use three gazetteers for person names, locations and organization names.[7] We semi-automatically enriched the location gazetteer using the Arabic Wikipedia[8] as well as other web sources. This enrichment consisted of: (i) taking the page labeled "*Countries of the world*" (دول العالم, dwl AlEAlm) as a starting point to crawl into Wikipedia and retrieve location names; (ii) we automatically filter the data removing stop words; (iii) finally, the resulting

[4] The authors did not report any per class comparison between SVM and CRF.

[5] http://chasen.org/∼taku/software/yamcha/

[6] http://crfpp.sourceforge.net/

[7] http://www.dsic.upv.es/∼ybenajiba

[8] http://ar.wikipedia.org

list goes through a manual validation step to ensure quality. On the training and test data, we tag only the entities which exist entirely in the gazetteer, e.g. if the entity 'United States of America' exists in our gazetteer, we would not tag 'United States' on the data as a location. Exception is made for person names. We augment our dictionary by converting the multiword names to their singleton counterparts in addition to keeping the multiword names in the list. We tag them on the evaluation data separately. Accordingly, the name 'Bill Clinton' and 'Michael Johnson' as two entries in our dictionary, are further broken down to 'Bill', 'Clinton', 'Michael', 'Johnson'. The intuition is that the system will be able to identify names such as 'Bill Johnson' and 'Clinton' as person names. This is always true for person names, however this assumption does not hold for location or organization names.

Part-Of-Speech (POS) tags and Base Phrase Chunks (BPC): To derive part of speech tags (POS) and base phrase chunks (BPC) for Arabic, we employ the AMIRA-1.0 system[9] described in (Diab et al., 2007). The POS tagger has a reported accuracy of 96.2% (25 tags) and the BPC system performs at a reported $F_{\beta=1}$=96.33%, assuming gold tokenization and POS tagging.

Nationality (NAT): The input is checked against a manually created list of nationalities.

Morphological features (MORPH): This feature set is based on exploiting the characteristic rich morphological features of the Arabic language. We rely on the MADA system for morphological disambiguation (Habash and Rambow, 2005), to extract relevant morphological information. MADA disambiguates words along 14 different morphological dimensions. It typically operates on untokenized texts (surface words as they naturally occur), hence, several of the features indicate whether there are clitics of different types. We use MADA for the preprocessing step of clitic tokenization (which addresses one of the challenges we note in Section 3, namely the impact different morphological surface forms have on sparseness). Recognizing the varying importance of the different morphological features and heeding the reported MADA performance per feature, we carefully engineered the choice of the relevant morphological features and their associated value representations. We selected 5 morphological features to include in this study.

1. Aspect (M_{ASP}) : In Arabic, a verb maybe imperfective, perfective or imperative. However since none of the NEs is verbal, we decided to turn this feature into a binary feature, namely indicating if a token is marked for Aspect (APP, for applicable) or not (NA, for not applicable).

2. Person (M_{PER}) : In Arabic, verbs, nouns, and pronouns typically indicate person information. The possible values are *first, second* or *third* person. Again, similar to *aspect*, the applicability of this feature to the NEs is more relevant than the actual value of *first* versus *second*, etc. Hence, we converted the values to APP and NA, where APP applies if the person feature is rendered as *first, second* or *third*.

3. Definiteness (M_{DEF}) : MADA indicates whether a token is definite or not. All the NEs by definition are definite. The possible values are DEF, INDEF or NA.

4. Gender (M_{GEN}) : All nominals in Arabic bear *gender* information. According to MADA, the possible values for this feature are masculine (MASC), feminine (FEM), and neuter (or not applicable NA), which is the case where gender is not applicable for instance in some of the closed class tokens such as prepositions, or in the case of verbs. We use the three possible values MASC, FEM and NA, for this feature. The intuition is that since we are using a sequence model, we are likely to see agreement in *gender* information in participants in the same NE.

5. Number (M_{NUM}) : For almost all the tokens categories (verbs, nouns, adjectives, etc.) MADA provides the grammatical *number*. In Arabic, the possible values are singular (SG), dual (DU) and plural (PL). The correlation of the SG value with most of the NEs classes is very high. Heeding the underlying agreement of words in Arabic when they are part of the same NE, the values for this feature are SG, DU, PL and NA (for cases where number is not applicable such as closed class function words).

Corresponding English Capitalization (CAP): MADA provides the English translation for the

[9] http://www1.cs.columbia.edu/∼mdiab/

words it morphologically disambiguates as it is based on an underlying bilingual lexicon. The intuition is that if the translation begins with a capital letter, then it is most probably a NE. This feature is an attempt to overcome the lack of capitalization for NEs in Arabic (see Section 3). This is similar to the *GlossCAP* feature used in (Farber et al., 2004).

4.3 Fuzzy Borda Voting Scheme

Fuzzy Borda Voting Scheme (FBVS) is useful when several possible candidates (c_n) are ranked by different experts (e_m) and we need to infer a single ranking (García Lapresta and Martínez Panero, 2002). It is based on the Borda count method which was introduced by Jean-Charles de Borda in 1770. In FBVS, each expert provides the ranking of the candidates with a weight[10] (w_n^m) assigned to each of them. Thereafter, for each expert e_i, we generate a square matrix such as $e_i = (r^i_{1,1} \ldots r^i_{n,n})$ where:

$$r^i_{j,k} = \frac{w^i_j}{w^i_j + w^i_k} \quad (1)$$

Given each expert matrix, we calculate for each row $r'^i_j = \sum_k r^i_{j,k};\ r^i_{j,k} > \alpha$ where α is a certain threshold. Accordingly, for each candidate, we sum up the weights obtained from the different experts in order to obtain a final weight for each candidate ($r''^j = \sum_i r'^i_j$). Finally, we rank them according to r''^j. In our experiments, the candidates we rank are the features. The FBVS ranking is calculated per ML technique and class of NEs across all the data sets according to the features' performances $F_{\beta=1}$, i.e. the weights. The $F_{\beta=1}$ ranges from $0-1$. We use $\alpha = 0.5$, thereby taking into consideration only the features which have shown a significant difference in performance.

5 Experiments and Results

5.1 Data

We report the results of our experiments on the standard sets of ACE 2003, ACE 2004 and ACE 2005 data sets.[11] The ACE data (see Table 1) is annotated for many tasks: Entity Detection and Tracking (EDT), Relation Detection and Recognition (RDR), Event Detection and Recognition (EDR). All the data sets comprise *Broadcast News* (BN) and *Newswire* (NW) genres. ACE 2004 includes an additional NW data set from the Arabic TreeBank (ATB). ACE 2005 includes a different genre of *Weblogs* (WL).

Corpus	*genre*	$Size_{train}$	$Size_{dev}$	$Size_{test}$
ACE 2003	BN	12.41k	4.12k	5.63k
	NW	23.85k	9.5k	9.1k
ACE 2004	BN	45.68k	14.44k	14.81k
	NW	45.66k	15.2k	16.9k
	ATB	19.04k	6.16k	6.08k
ACE 2005	BN	18.54k	5k	8.4k
	NW	40.26k	12.5k	13.83k
	WL	13.7k	6.2k	6.4

Table 1: Statistics of ACE 2003, 2004 and 2005 data

We create a dev, test and train set for each of the collections. Table 1 gives the relevant statistics. It is worth noting that the standard training sets have 4 folds that are typically used for training. We used one of the folds as dev data for tuning purposes, rendering our training data less for our experiments. For data preprocessing, we remove all annotations which are not oriented to the EDR task. Also, we remove all the 'nominal' and 'pronominal' mentions of the entities and keep only the 'named' ones. Hence, all the listed characteristics for this corpus pertain to the portions of the data that are relevant to NER only. The ACE 2003 data defines four different NE classes: Person (e.g. Albert Einstein), Geographical and Political Entities (GPE) (e.g. Kazakhistan), Organization (e.g. Google Co.) and Facility (e.g. the White House). Whereas in ACE 2004 and 2005, two NE classes are added to the ACE 2003 tag-set: Vehicles (e.g. Rotterdam Ship) and Weapons (e.g. Kalashnikof). In order to overcome the sparseness issues resulting , we clitic tokenize the text using the MADA system. We use the ATB style clitic tokenization standard. Finally, we convert the data from the ACE format into the IOB2 annotation scheme (Tjong Kim Sang and De Meudler, 2003).

5.2 Experimentation

Our objective is to find the optimum set of features per NE class and then combine the outcome in a

[10] weights are not required for classical Borda count.

[11] http://www.nist.gov/speech/tests/ace/

global NER system for Arabic. We set the context window to be of size $-1/+1$ for all the experiments, as it empirically yields the best performance. We use the CoNLL evaluation metrics of precision, recall, and $F_{\beta=1}$ measures. The CoNLL metrics are geared to the chunk level yielding results as they pertain to the entire NE (Tjong Kim Sang and De Meudler, 2003). Our experiments are presented as follows:

1. Training per individual NE class: We train for an individual class by turning off the other annotations for the other classes in the training set. We experimented with two settings: 1. Setting all the other NE classes to O, similar to non-NE words, thereby yielding a 3-way classification, namely, B-NE and I-NE for the class of interest, and O for the rest including the rest of the NEs and other words and punctuation; 2. The second setting discriminated between the other NE classes that are not of interest and the rest of the words. The intuition in this case is that NE class words will naturally behave differently than the rest of the words in the data. Thereby, this setting yields a 4-way classification: B-NE and I-NE for class of interest, NE for the other NE classes, and O for the other words and punctuation in the data. In order to contrast the 3-way vs the 4-way classification, we run experiments and evaluate using the ACE 2003 data set with no features apart from 'CXT' and 'current word' using SVM. Table 2 illustrates the yielded results. For all

Class	*Num(classes)*	*BN genre*	*NW genre*
GPE	3	76.72	79.88
	4	**76.88**	**80.99**
PER	3	64.34	42.93
	4	**67.56**	**44.43**
ORG	3	41.73	25.24
	4	**46.02**	**37.97**
FAC	3	**23.33**	15.3
	4	**23.33**	**18.12**

Table 2: $F_{\beta=1}$ Results using 3-way vs. 4-way class annotations using SVM

the NE classes we note that the 4-way classification yields the best results. Moreover, we counted the number of 'conflicts' obtained for each NE classification. A 'conflict' arises when the same token is classified as a different NE class by more than one classification system. Our findings are summarized as follows:

(i). 3 classes: 16 conflicts (8 conflicts in BN and 8 in NW). 10 of these conflicts are between GPE and PER, and 6 of them are between GPE and ORG.

(ii). 4 classes: 10 conflicts (3 conflicts in BN and 7 in NW). 9 of these conflicts are between GPE and ORG, and only one of them is between GPE and FAC.

An example of a conflict observed using the 3-way classification that disappeared when we apply the 4-way classification is in the following sentence: نشرت صحيفة واشنطن تايمس تقريرا *n$rt SHyfp WA$nTn tAyms tqryrA*, which is translated as 'The Washington Times newspaper published a report'. When trained using a 3-way classifier, 'Washington' is assigned the tag GPE by the GPE classifier system and as an ORG by the ORG classifier system. However, when trained using the 4-way classifier, this conflict is resolved as an ORG in the ORG classifier system and an NE in the GPE classifier system. Thereby confirming our intuition that a 4-way classification is better suited for the individual NE classification systems. Accordingly, for the rest of the experiments in this paper reporting on individual NE classifiers systems, we use a 4-way classification approach.

2. Measuring the impact of Individual features per class : An experiment is run for each fold of the data. We train on data annotated for one NE class, one Machine Learning (ML) method (i.e. SVM or CRF), and one feature. For each experiment we use the tuning set for evaluation, i.e. obtaining the $F_{\beta=1}$ performance value.

3. FBVS Ranking : After obtaining the F-measures for all the individual features on all the data genres and using the two ML techniques, we rank the features (in a decreasing order) according to their impact (F-measure obtained) using FBVS (see 4.3). This results in a ranked list of features for each ML approach and data genre per class. Once the features are ranked, we incrementally experiment with the features in the order of the ranking, i.e. train with the first feature and measure the performance on the tuning data, then train with the second together with the first feature, i.e. the first two features and measure performance, then the first three features and so on.

Feats	PER	GPE	ORG	FAC	VEH/WEA
LEX_1	16	12	12	15	4
LEX_2	3	15	7	12	5
LEX_3	10	6	15	10	6
LEX_4	7	16	4	8	7
LEX_5	15	14	16	16	8
LEX_6	12	4	10	9	9
GAZ	14	7	9	11	3
BPC	4	13	13	6	1
POS	1	5	1	4	16
NAT	8	3	2	3	15
M_{ASP}	13	2	5	2	10
M_{PER}	11	11	3	5	14
M_{DEF}	9	9	6	7	11
M_{GEN}	5	8	11	13	12
M_{NUM}	6	10	14	14	13
CAP	2	1	8	1	2

Table 3: Ranked features according to FBVS using SVM for each NE class

4. Feature set/class generalization : Finally, we pick the first *n* features that yield the best converging performance (after which additional features do not impact performance or cause it to deteriorate). We use the top *n* features to tag the test data and compare the results against the system when it is trained on the whole feature set.

5.3 Individual Features Experiments

After running experiments using each feature individually, each result is considered an expert (the obtained F-measure is the weight in this framework).

Our goal is to find a general ranking of the features for each ML approach and each class. Table 3 shows the obtained rankings of the features for each class using SVM. It is worth noting that the obtained CRF rankings are very similar to those yielded by using SVM. We note that there are no specific features that have proven to be useless for all classes and ML approaches.

5.4 Feature set/class Experiments

We combine the features per NE class incrementally. Since the total number of features is 16, each ML classifier is trained and evaluated on the tuning data 16 times for each genre. A best number of features per class per genre per ML technique is determined based on the highest yielded $F_{\beta=1}$. Finally, the last step is combining the outputs of the different classifiers for all the classes. In case of conflicts, where the same token is tagged as two different NE classes, we use a simple heuristic based on the classifier precision for that specific tag, favoring the tag with the highest precision.

Table 4 illustrates the obtained results. For each data set and each genre it shows the F-measure obtained using the best feature set and ML approach. We show results for both the dev and test data using the optimal number of features **Best Feat-Set/ML** contrasted against the system when using all 16 features per class **All Feats/ML**. The table also illustrates three baseline results on the test data only. **FreqBaseline**: For this baseline, we assign a test token the most frequent tag observed for it in the training data, if a test token is not observed in the training data, it is assigned the most frequent tag which is the O tag. **MLBaseline**: In this baseline setting, we train an NER system with the full 16 features for all the NE classes at once. We use the two different ML approaches yielding two baselines: **MLBaseline**$_{SVM}$ and **MLBaseline**$_{CRF}$.
It is important to note the difference between the **All Feats/ML** setting and the **MLBaseline** setting. In the former, **All Feats/ML**, all 16 features are used per class in a 4-way classifier system and then the classifications are combined and the conflicts are resolved using our simple heuristic while in the latter case of **MLBaseline** the classes are trained together with all 16 features for all classes in one system. Since different feature-sets and different ML approaches are used and combined for each experiment, it is not possible to present the number of features used in each experiment in Table 4. However, Table 5 shows the number of features and the ML approach used for each genre and NE class.

6 Discussion and Error Analysis

As illustrated in Table 5, SVM outperformed CRF on most of the classes. Interestingly, CRF tends to model the ORG and FAC entities better than SVM. Hence, it is not possible to give a final word on the superiority of SVM or CRF in the NER task, and it is necessary to conduct a per class study, as the one we present in this paper, in order to determine the right ML approach and features to use for each class. Therefore, our best global NER system combined

		ACE 2003		*ACE 2004*			*ACE 2005*		
		BN	*NW*	*BN*	*NW*	*ATB*	*BN*	*NW*	*WL*
	FreqBaseline	73.74	67.61	62.17	51.67	62.94	70.18	57.17	27.66
	MLBaseline$_{SVM}$	80.58	76.37	74.21	71.11	73.14	79.3	73.9	54.68
	MLBaseline$_{CRF}$	81.02	76.18	74.67	71.8	73.04	80.13	74.75	55.32
dev	**Best Feat-set/ML**	**83.41**	**79.11**	**76.9**	**72.9**	**74.82**	**81.42**	**76.07**	**54.49**
	All Feats. SVM	81.79	77.99	75.49	71.8	73.71	80.87	75.69	53.73
	All Feats. CRF	81.76	76.6	76.26	71.85	74.19	79.66	74.83	36.11
test	**Best Feat-set/ML**	**83.5**	**78.9**	**76.7**	**72.4**	**73.5**	**81.31**	**75.3**	**57.3**
	All Feats. SVM	81.76	77.27	74.71	71.16	73.63	81.1	72.41	55.58
	All Feats. CRF	81.37	75.89	75.73	72.36	74.21	80.16	74.43	27.36

Table 4: Final Results obtained with selected features contrasted against all features combined

	BN		*NW*		*ATB*		*WL*	
	N	*ML*	*N*	*ML*	*N*	*ML*	*N*	*ML*
Person	12	SVM	14	SVM	9	SVM	11	SVM
Location	10	SVM	7	SVM	16	CRF	14	SVM
Organization	9	CRF	6	CRF	10	CRF	12	CRF
Facility	10	CRF	14	CRF	14	SVM	16	CRF
Vehicle	3	SVM	3	SVM	3	SVM	3	SVM
Weapon	3	SVM	3	SVM	3	SVM	3	SVM

Table 5: Number of features and ML approach used to obtain the best results

the results obtained from both ML approaches.

Table 4, shows that our **Best Feat-set/ML** setting outperforms the baselines and the **All Feats** {**SVM/CRF**} settings for all the data genres and sets forthe test data. Moreover, the **Best Feat-set/ML** setting outperforms both **All Feats** {**SVM/CRF**} settings for the dev data for all genres except for ACE2003 NW, where the difference is very small.

The results yielded from the ML baselines are comparable across all the data genres and the two ML approaches.

Comparing the global ML baseline systems against the All Feature Setting, we see that the **All Feats** setting consistently outperforms the **MLBaseline** settings except for ACE2005 NW data set. This suggests that training separate systems for the different NEs has some benefit over training in one global system.

Comparing the performance per genre across the different data sets. We note better performance across the board for BN data over NW per year. The worst results are yielded for ACE 2004 data for both BN and NW genres. There is no definitive conclusion that a specific ML approach is better suited for a specific data genre. We observe slightly better performance for the CRF ML approach in the **MLBaseline**$_{CRF}$ condition for both BN and NW.

The worst performance is yielded for the WL data. This may be attributed to the small amount of training data available for this genre. Moreover the quality of the performance of the different feature extraction tools such as AMIRA (for POS tagging and BPC) and MADA (for the morphological features) are optimized for NW data genres, thereby yielding suboptimal performance on the WL genre, leading to more noise than signal for training. However, comparing relative performance on this genre, we see a significant jump from the most frequent baseline **FreqBaseline** ($F_{\beta=1}$=27.66) to the best baseline **MLBaseline**$_{CRF}$ ($F_{\beta=1}$=55.32). We see a further significant improvement when the **Best Feat-set/ML** setting is applied yielding an $F_{\beta=1}$=57.3. Interestingly, however the **MLBaseline**$_{CRF}$ yields a much better performance ($F_{\beta=1}$=55.32) than **All Feats CRF** with an $F_{\beta=1}$=27.36. This may indicate that a global system that trains all classes at once using CRF for sparse data is better than training separate classifiers and then combining the out-

puts. It is worth noting the difference between **MLBaseline**$_{SVM}$ and **All Feats SVM**, $F_{\beta=1}$=54.68 and $F_{\beta=1}$=55.58, respectively. This result suggests that SVM are more robust to less training data as illustrated in the case of the individual classifiers in the latter setting.

Comparing dev and test performance, we note that the overall results on the dev data are better than those obtained on the test data, which is expected given that the weights for the FBVS ranking are derived based on the dev data used as a tuning set. The only counter example for this trend is with the WL data genre, where the test data yields a significantly higher performance for all the conditions except for **All Feats CRF**.

As observed in Table 3, the ranking of the individual features could be very different for two NE classes. For instance, the BPC is ranked 4th for the PER class and is ranked 13th for GPE and ORG classes. The disparity in ranking for the same individual features strongly suggests that using the same features for all the classes cannot lead to a global optimal classifier. With regards to morphological features, we note in Table 3, that Definiteness, M_{DEF}, is helpful for all the NE classification systems, by virtue of being included for all optimal systems for all NE classification systems. Aspect, M_{ASP}, is useful for all classes except PER. Moreover, M_{GEN} and M_{NUM}, corresponding to Gender and Number, respectively, contributed significantly to the increase in recall for PER and GPE classes. Finally, the Person feature, M_{PER} contributed mostly to improving the classification of ORG and FAC classes. Accordingly, observing these results, contrary to previous results by (Farber et al., 2004), our results strongly suggest the significant impact morphological features have on Arabic NER, if applied at the right level of granularity.

Inconsistencies in the data lead to many of the observed errors. The problem is that the ACE data is annotated primarily for a mention detection task which leads to the same exact words not being annotated consistently. For instance, the word 'Palestinians' would sometimes be annotated as a GPE class while in similar other contexts it is not annotated as a named entity at all. Since we did not manually correct these cases, the classifiers are left with mixed signals. The VEH and WEA classes both exhibit a uniform ranking for all the features and yield a very low performance. This is mainly attributed to the fact that they appear very rarely in the training data. For instance, in the ACE 2005, BN genre, there are 1707 instances of the class PER, 1777 of GPE, 103 of ORG, 106 of FAC and only 4 for WEA and 24 for VEH.

7 Conclusions and Future Directions

We described the performance yielded using language-dependent and language independent features in SVM and CRF for the NER task on different standard Arabic data-sets comprising different genres. We have measured the impact of each feature individually on each class, we ranked them according to their impact using the Fuzzy Borda Voting Scheme, and then performed an incremental features' selection considering each time the N best features.

We reported the importance of each feature for each class and then the performance obtained when the best feature-set is used. Our experiments yield state of the art performance significantly outperforming the baseline. Our best results achieve an $F_{\beta=1}$ score of 83.5 for the ACE 2003 BN data. Our ACE2005 results are state of the art when compared to the best system to date. It is worth noting that these obtained results are trained on less data since we train only on 3 folds vs the standard 4 folds. Our results show that the SVM and CRF have very similar behaviors. However, SVM showed more robust performance in our system using data with very random contexts, namely for the WL data, i.e. Weblogs. We definitively illustrate that correctly exploiting morphological features for languages with rich morphological structures yields state of the art performance. For future work, we intend to investigate the use of automatic feature selection methods on the same data.

Acknowledgments The authors would like to thank the reviewers for their detailed constructive comments. We would like to thank MCyT TIN2006-15265-C06-04 and PCI-AECI A/010317/07 research projects for partially funding this work. Mona Diab would like to acknowledge DARPA GALE Grant Contract No. HR0011-06-C-0023 for partially funding this work.

References

Bogdan Babych and Anthony Hartley. 2003. *Improving Machine Translation Quality with Automatic Named Entity Recognition*. In *Proc. of EACL-EAMT*. Budapest.

Yassine Benajiba and Paolo Rosso. 2008. *Arabic Named Entity Recognition using Conditional Random Fields*. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC'08*.

Yassine Benajiba, Paolo Rosso and José Miguel Benedí. 2007. *ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy*. In *Proc. of CICLing-2007*, Springer-Verlag, LNCS(4394), pp. 143-153.

Yassine Benajiba and Paolo Rosso. 2007. *ANERsys 2.0 : Conquering the NER task for the Arabic language by combining the Maximum Entropy with POS-tag information*. In *Proc. of Workshop on Natural Language-Independent Engineering, IICAI-2007*.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. *Maximum Entropy Models For Named Entity Recognition*. In *Proc. of CoNLL-2003*. Edmonton, Canada.

Tim Buckwalter. 2002. *Buckwalter Arabic Morphological Analyzer*. In *Linguistic Data Consortium. (LDC2002L49)*.

Mona Diab, Kadri Hacioglu and Daniel Jurafsky. 2007. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, chapter 9, pp. 159–179. Abdelhadi Soudi, Antal van den Bosch and Gunter Neumann (Eds.), Springer.

Benjamin Farber, Dayne Freitag, Nizar Habash and Owen Rambow. 2008. *Improving NER in Arabic Using a Morphological Tagger.* In *Proc. of LREC'08*.

Sergio Ferrández, Óscar Ferrández, Antonio Ferrández and Rafael Muñoz. 2007. *The Importance of Named Entities in Cross-Lingual Question Answering* In *Proc. of RANLP'07*.

José Luis García Lapresta and Miguel Martínez Panero 2002. *Borda Count Versus Approval Voting: A Fuzzy Approach*. Public Choice, 112(1-2):pp. 167–184.

Nizar Habash and Owen Rambow. 2005. *Arabic Tokenization, Part-Of-Speech Tagging and Morphological Disambiguation in One Fell Swoop*. In *Proc. of Workshop of Computational Approaches to Semitic Languages, ACL-2005*.

John Lafferty, Andrew McCallum and Fernando Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In *Proc. of ICML-2001*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. In *Proc. of CoNLL-2003*. pp. 142–147.

Hiroyuki Toda and Ryoji Kataoka. 2005. *A Search Result Clustering Method using Informatively Named Entities.*. In *Proc. of the 7th ACM International Workshop on Web Information and Data Management*.

Q. Tri Tran, T.X. Thao Pham, Q. Hung Ngo,Dien Dinh, and Nigel Collier. 2007. *Named Entity Recognition in Vietnamese documents*. *Progress in Informatics Journal*. 2007.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. *Springer Verlag*.