An empirical Comparison of Arabic Named Entity Recognition Methods: Application to the ALP Corpus

Mohamed Lichouri
USTHB, Algiers, Algeria
mlichouri@usthb.dz

Abstract

This study compares the performance of some existing approaches to the problem of Arabic Named Entity Recognition. The approaches under consideration are based on Sequence Labelling and Multi-Label Classification methods. We will use the ALP corpus, a newly produced corpus with more than 58 tags, as our single corpus for comparison in order to ensure a fair comparison. In other words, we'll use a 58-way categorization procedure to figure out what each token's tags are. Despite just employing a portion of the ALP corpus—ALP2 (50%) and ALP3 (25%)—an average accuracy of more than 88% was achieved, which make the results highly encouraging.

1 Introduction

Named Entity Recognition (NER) focuses on the challenge of identifying specific linguistic categories that share semantic characteristics, such as organization names, where despite their outward variances, they all communicate the same meaning. Furthermore, they commonly emerge in environments that are similar. Similar rules apply to names of individuals, places, or dates. Sometimes people think that the NER problem has been resolved. We can say that well-trained systems score almost as high as human performance, at the very least. Neural networks, rules-based systems, and statistical models like CRFs and Maximum Entropy have all been used to close the efficiency gap with humans. Consequently, why bother with it? Because NER today has more to do with data than it does with algorithms (Frederic Giannetti, 2018). This is true for high resourced and low resource languages such as Arabic. This is why, in this paper, we will highlight the important work done on Arabic Named Entity Recognition. In overall there is so much progress for NER in other language like English, German and French, as opposed to the Arabic Language. The complexity of the Arabic language, peculiarities in the Arabic orthographic system, non-standardization of the written text, ambiguity, and lack of resources are the main reasons for the minimum number of research in NER.

Another constraint is the non conformity between the different tagging model, where some adopt the rule token from foreign languages and applied to Arabic, whereas other like Abed Alhakim Freihat opted to create a more thoroughly list of tags which can express the maximum number and variation of the Arabic language. This is why in this paper, we have considered the corpus created by Abed Alhakim Freihat as a test ground. So the novelty of this paper relate to:

- The first use of a mega corpus (ALP) (Freihat et al., 2018a,b) that contains more than 2 millions tagged word.
- The first ever conduction of a 58-way classification in Arabic (to our knowledge).
- Conducting a comparison study between some existing approaches using some well known tools for NER.

The rest of the paper will be organised as follow. An extensive and exhaustive list of work have been presented as a reference in the section 2. In section 3, we will present a description of the used dataset, followed by the different used approaches in section 4 as well as the gotten results in section 5. Whereas we will conclude our paper in section 6.

2 Related Work

The first work (to our knowledge) on Arabic Named Entity Recognition (ANER) was done by Benajiba et al. (Benajiba et al., 2007), where they first build an ANER system for Arabic texts based-on n-grams and maximum entropy which is applied to their own training and test corpora (ANERcorp)

and gazetteers (ANERgazet). An overall accuracy of 55.23% was achieved by this first experiment, which was further improved by 19 point by the same authors in their second work (Benajiba and Rosso, 2008) by using additional information such as Part-Of-Speech tags and Base Phrase Chunks and changing the probabilistic model from Maximum Entropy to Conditional Random Fields. Another ANER system was built by Shaalan and Raza (Shaalan and Raza, 2009) using a rule-based approach. The process used by the authors is as follow: (a) recognizing the named entities by using a Whitelist which is representing a dictionary of names, and a grammar, in the form of regular expressions then (b) applying a filtration mechanism to revise the gotten results in (a) by using metadata and also a Blacklist or rejecter for case of ill-formed named entities and last (c) a disambiguation of identical or overlapping textual matches returned by different name entity extractors to get the correct choice. NERA has achieved an average accuracy of over 80% for the 10 used NEs tags. An improvement of the coverage of the mis-classified person, location and organization named entities types by 69.93 per cent, 57.09 per cent and 54.28 per cent, respectively was achieved by NERA 2.0 by the same authors (Oudah and Shaalan, 2017) by following an hybrid approach that integrates both rule-based and machine learning-based NER approaches. By incorporating cross-lingual features and knowledge bases from English using cross-lingual links, Darwish (Darwish, 2013) show that such features have a dramatic positive effect on recall where the effectiveness of cross-lingual features and resources on a standard dataset has permit the author to achieve a relative improvement of 4.1% over the best reported result in the literature. In recent year, we note the work done by Lample et al.(Lample et al., 2016) where they introduce two new neural architectures—one based on bidirectional LSTMs and conditional random fields, and the other that constructs and labels segments using a transition-based approach inspired by shift-reduce parsers. The authors consider also that character-based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora are considered as two sources of information about words in their model.An overall accuracy of over 78% was obtained in NER in four languages (English, Spanich, German and Ducth) without resorting to any language-specific knowledge or resources such as gazetteers. There is also the work of Lhioui et al.(Lhioui et al., 2017) where they used the NooJ platform based on linguistic rules to manage an experiments on the pilot Arabic Propbank data to finally achieve a score of 87%, which they proclaim that improves the current state of the art in Arabic NE recognition. Where-as Elbazi and Laachfoubi (El Bazi and Laachfoubi, 2017) have introduced a features based on Latent Dirichlet Allocation (LDA) to investigate and analyze three different approaches for utilizing LDA, Topical Prototypes approach and Topical Word Embeddings approach. The authors proclaim that their experiments show that each of the presented approaches improves the baseline features, among which the Word-Class LDA approach performs the best (over 73%). Moreover, the combination of these topic modeling approaches provides additive improvements, outperforming traditional word representations as Skip-gram word embeddings and Brown Clustering. The same authors (Bazi and Laachfoubi, 2018) have recently investigated whether word representations can also boost supervised NER in Arabic by using word representations as additional features in a Conditional Random Field (CRF) model and compare in the same time three neural word embedding algorithms (SKIP-gram, CBOW and GloVe) and six different approaches for integrating word representations into NER system where the Brown Clustering achieved the best performance among the six approaches by an accuracy of 67%.

Corpus	**ALP2** (50%)	**ALP3** (25%)	**ALP**
# tokens	1.04M	524.28k	2.27M
# unique tokens	84.13k	64k	148k
# labels	1.04M	524.28k	2.27M
# unique labels	54	50	58

Table 1: ALP corpus statistics

3 Dataset

In this work, we used the ALP corpus (Freihat et al., 2018a,b). The whole corpus had been tokenized and tagged in a semi-supervised way, where the authors started by labeling a 200 tokens and used it as training to predict the tags of another set of 200 tokens. The resulted tags have been verified manually by an expert which resulted in a 400 tokens as a training dataset. The authors have repeated this process until they created this ALP corpus, which

contain more than 2 millions fully tagged tokens. For this work we have divided this corpus to two sets, ALP2 and ALP3 sets. In table 1, we provide some statistics on the used corpora.

In the table 3, we will present the labels frequency in the total corpus.

Label	Frequency	Example
O	2069010	الاسمنت
B-LOC	42972	الكعبة
I-ORG	31537	الفلكي
B-PER	28247	يوسف
B-ORG	20826	لجنة
I-PER	20109	الباشا
I-LOC	19964	المتحدة
C+B-LOC	13128	ودمشق
B-MONTH	6581	سبتمبر
P+B-LOC	4947	بفيينا
P+B-ORG	2851	للجزيرة
B-DAY	2267	الاثنين
I-EVENT	2173	اليمن
ALLAH	1875	الله
B-EVENT	1424	قمة
C+P+B-LOC	1315	وبالسودان
B-MISC	1298	إيرباص
C+B-PER	1220	وضياء
I-MONTH	1112	الأول
C+B-ORG	968	والاتحاد
P+B-PER	768	لمحمد
I-MISC	621	رختر
B-CLAN	434	الروهينغا
I-AWARD	206	سلطان
B-TIME	197	الساعة
I-CLAN	192	العربية
P+B-EVENT	151	لمهرجان
I-TIME	138	العاشرة
C+P+ALLAH	114	ولله

Table 2: ALP corpus labels frequency and examples -Part 1-

Label	Frequency	Example
B-AWARD	105	نوبل
P+ALLAH	104	لله
I-PROPH	104	محمد
C+B-MISC	91	وأندروميدا
C+B-CLAN	73	وآل
P+B-MISC	67	للمريخ
C+ALLAH	59	والله
C+B-MONTH	43	وجمادى
C+B-EVENT	38	وحرب
C+B-DAY	25	وخميس
ALLAH+VOC	20	اللهم
P+B-CLAN	15	لبني
C+P+B-PER	15	ولابن
C+B-AWARD	15	وجائزة
P+B-PROPH	10	للرسول
P+B-AWARD	8	لجائزة
C+P+B-ORG	2	وللأمم
B-CHAPTER	2	الفاتحة
C+P+B-PROPH	1	وللرسول
B-ORH	1	كيف
C+B-TIME	1	والثالثة

Table 3: ALP corpus labels frequency and examples -Part 2-

4 Approaches

We will present in this section our two proposed approach where the first one is our proposed approach which is based on Multi-Label Classification technique whereas the second is the Sequence Labeling approach.

4.1 Multi-Label Classification Approach

In this approach, we address the problem of NER as a simple Multi-Label Classification problem. Where the labels in the used corpus are considered as class candidate. For example if we have 5 label, the classification will be a 5-way classification approach. The following algorithm (–see algorithm 1)will summarize the different step for this approaches.

Algorithm 1 Multi-Label Classification

procedure MULTI-LABELCLASSIFICATION($corpus$)
    Preparing Train and Test Data ▷ (Step 1)
    Convert Train and Test Data to array ▷ (Step 2)
    Applying TFidf transformation
    Training Phase for LSVM, BNB, MNB, LR, SGD and PAC ▷ (Step 3)
    for $W \in Test$ **do** ▷ Testing Phase (Step 4)
        Predicting the Class of W by the six classifier

4.2 Sequence Labelling Approach

When the aim of NER is to extract the name of country, person in a text, we can note that the human being, when reading a news article he would usually recognise that a word or a phrase refers to a country, a person name, even when he has not seen that name before. The main reason is that there are many different cues in the sentence or the whole article that can be used to determine whether a word or a phrase is a country name or person name. This is where this approach perform well, because it take advantage of the surrounding context when labelling tokens in a sequence, where a commonly used method is the conditional random field (CRF). Which is a type of probabilistic graphical model that can be used to model sequential data, such as labels of words in a sentence.

In CRF, a set of feature functions, will be designed to extract features for each word in a sentence. During model training, CRF will try to determine the weights of different feature functions that will maximise the likelihood of the labels in the training data.

In the following algorithm 2, we will present the main steps for sequence labeling a word in a sentence.

Algorithm 2 Sequence Labeling

procedure SEQUENCE LABELING($corpus$)
    Generating Part-of-Speech Tags ▷ (Step 1)
    for $W \in corpus$ **do** ▷ Generating Word Features (Step 2)
        f1 := Convert W[i] to lower case
        f2 := Prefix/Suffix of W[i]
        f3 := W[i-1] (previous), W[i+1] (next)
        f4 := if(W[i]) is Uppercase or Lowercase (1 or 0)
        f5 := if(W[i]) is Number or Contains digit (1 or 0)
        f6 := PosTag(W[i]), PosTag(W[i-1]), PosTag(W[i+1])
        f7 := if(W[i]) contains special character (1 or 0)
    Split to train and test set ▷ (Step 3)
    Train CRF Model ▷ (Step 4)
    for $W \in test$ **do** ▷ Testing phase (Step 5)
        Predict the tag of W[i] by CRF.tagger

5 NER Experiment Setup and Result

Because the ALP corpus has a huge number of instance, we couldn't conduct the desired experiments, this is why we decided to use only the half of the corpus, which give use slightly more than 1 Million labeled token, lets name it **ALP2**.

5.1 Multi-Label Classification Experiments

We considered a set machine learning techniques using the scikit-learn library (Pedregosa et al., 2011), namely: Support Vector Machines (SVM), Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB), Logistic Regression (LR), Stochastic Gradient Descent (SGD) and Passive Aggressive (PAC). For this classifiers we opted for the default configuration as in the scikit-learn.

	Precision	**Recall**	**F1-Score**	**Accuracy**
LSVC	81%	87%	83%	86,73%
BNB	66%	81%	73%	81.27%
MNB	78%	84%	79%	84.40%
LogReg	79%	86%	81%	85.67%
SGD	76%	83%	76%	82.81%
PAC	80%	86%	83%	86.15%

Table 4: Detailed Results on a non shuffled dataset. Precision, Recall and F1-score are in average mode.

	Precision	**Recall**	**F1-Score**	**Accuracy**
LSVC	85%	90%	87%	90.38%
BNB	76%	86%	80%	86.45%
MNB	82%	88%	84%	88.46%
LogReg	84%	90%	86%	89.54%
SGD	82%	88%	83%	87.93%
PAC	84%	90%	87%	89.81%

Table 5: Detailed Results on a shuffled dataset. Precision, Recall and F1-score are in average mode.

For this approach, we carried-out two experiments: the first without shuffling the data (see table 4), when splitting the corpus to train and test.

Where-as the second by shuffling the data (see table 5). As mentioned earlier, we took only half of the ALP corpus, with a size of 1.04 Million tokens (ALP2). We divided this corpus to a 80% for train and the rest for test. For this approach, the best results has been gotten by the **LSVC** classifier when shuffling the data with an average accuracy of 90.38%.

	Setup	Accuracy
w/o Pos-Tags	ALP3	100%
	ALP2	99.9%
+ Pos-Tags	ALP3	90.1%
	ALP2	87.1%

Table 6: Accuracy gotten with sklearn-crf.

5.2 Sequence Labeling Classification Experiments

We used the code in [1] by Francois Vanderseypen. This tools is based on the sklearn_crfsuite[2],which permit to label a sequence of word with or without using Pos-Tags information. This is why we conducted four experiments: two with Pos-Tags and two without Pos-Tags using different setups. The gotten results as well as a description of the used dataset is described in table 6. We should note that we used for once 50% of the ALP (let's name it ALP2) and for the second 25% of ALP (let's name it ALP3). This choice was made because of the lack of computing power.

If we consider the same setup as for the first Approach, while using the ALP2 corpus, the best results achieved by this approach is with a an accuracy of 99.9% without using the Pos-Tags. Whereas, while using the ALP3 corpus, a perfect accuracy was obtained without using Pos-Tags. If we consider the Pos-Tags information, we noted a decrease of about 10% in accuracy.

6 Conclusion

We presented in this paper an empirical comparison between two approaches and two tools. Where the first approach is based on a Multi-Label Classification Methods and the second approach is based on a sequence labeling methods (two tools). For the Multi-Label Classification, the best results was achieved by LSVM with an accuracy of 90.38%, which is very encouraging because the time of training is very low in comparison to the other tool. Or the tool, which is based on sklearn-crf has achieved some excellent results, despite the very long training time.

[1]https://github.com/Orbifold/dutch-ner

[2]https://github.com/TeamHG-Memex/sklearn-crfsuite

References

Frederic Giannetti. 2018. Named Entity Recognition: Challenges and Solutions. https://blog.doculayer.com/named-entity-recognition-challenges\-and-solutions. Online; accessed 18 December 2022.

Ismail El Bazi and Nabil Laachfoubi. 2018. Arabic named entity recognition using word representations. *arXiv preprint arXiv:1804.05630*.

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*, volume 8, pages 143–153. Citeseer.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 143–153. Springer.

Kareem Darwish. 2013. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1558–1567.

Ismail El Bazi and Nabil Laachfoubi. 2017. Arabic named entity recognition using topic modeling. *context*, 230.

Abed Alhakim Freihat, Mourad Abbas, Gábor Bella, and Fausto Giunchiglia. 2018a. Towards an optimal solution to lemmatization in arabic. In *Fourth International Conference On Arabic Computational Linguistics, ACLING 2018, November 17-19, 2018, Dubai, United Arab Emirates*, volume 142 of *Procedia Computer Science*, pages 132–140. Elsevier.

Abed Alhakim Freihat, Gabor Bella, Hamdy Mubarak, and Fausto Giunchiglia. 2018b. A single-model approach for arabic segmentation, pos tagging, and named entity recognition. In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–8. IEEE.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Chahira Lhioui, Anis Zouaghi, and Mounir Zrigui. 2017. A rule-based approach for arabic temporal expression extraction. In *2017 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE.

Mai Oudah and Khaled Shaalan. 2017. Nera 2.0: Improving coverage and performance of rule-based named entity recognition for arabic. *Natural Language Engineering*, 23(3):441–472.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Khaled Shaalan and Hafsa Raza. 2009. Nera: Named entity recognition for arabic. *Journal of the American Society for Information Science and Technology*, 60(8):1652–1663.