Mapping Arabic Wikipedia into the Named Entities Taxonomy

Fahd Alotaibi and Mark Lee
School of Computer Science, University of Birmingham, UK
{fsa081|m.g.lee}@cs.bham.ac.uk

ABSTRACT

This paper describes a comprehensive set of experiments conducted in order to classify Arabic Wikipedia articles into predefined sets of Named Entity classes. We tackle using four different classifiers, namely: Naïve Bayes, Multinomial Naïve Bayes, Support Vector Machines, and Stochastic Gradient Descent. We report on several aspects related to classification models in the sense of feature representation, feature set and statistical modelling. The results reported show that, we are able to correctly classify the articles with scores of 90% on Precision, Recall and balanced F-measure.

KEYWORDS: Arabic Named Entity, Wikipedia, Arabic Document Classification, Supervised Machine Learning.

Proceedings of COLING 2012: Posters, pages 43–52,
COLING 2012, Mumbai, December 2012.

1 Introduction

Relying on supervised machine learning technologies to recognize Named Entities (NE) in the text requires the development of a reasonable volume of data for the training phase. Manually developing a training dataset that goes beyond the news-wire domain is a non-trivial task.

Examination of online and freely available resources, such as the Arabic Wikipedia (AW) offers promise because the underlying scheme of AW can be exploited in order to automatically identify NEs in context. To utilize this resource and develop a NEs corpus from AW means two different tasks should be addressed: 1) Identifying the NEs in the context regardless of assigning those into NEs semantic classes. 2) Classifying AW articles into predefined NEs taxonomy.

The first task has already been addressed in Alotaibi and Lee (2012) where they present a novel approach to identify the NEs in AW by transforming the news-wire domain to facilitate binary NEs classification and to extract contextual and language-specific features, which are then compiled into a classifier.

In this study we investigated the problem of classifying AW articles into NEs categories, exploiting both the Wikipedia-specific format and Arabic language features. We modelled this problem as a document classification task in order to assign each AW article into a particular NEs class. We decided to apply the coarse-grained NEs classes provided by ACE (2008).

After conducting a comprehensive set of experiments, we were able to identify the three-tuples {Feature representation, Features set, Statistical model} for best performance. We found that the 3-tuples {TF-IDF, FF, SGD} gave the highest results with scores of 90% in all metrics.

2 Mapping Wikipedia into NEs Taxonomy

2.1 Selecting Named Entities Classes

For the purpose of this study, we decided to adopt the ACE (2008) taxonomy of named entities for our corpus. However, some ACE (2008) classes required slight amendments in order to be better suited for use in an open domain corpus, such as Wikipedia. For example, we found that there are many articles in Wikipedia related to products and therefore, we decided to add a "Product" class. In addition, we used a "Not-Named-Entity" class to indicate that the article does not reference a named entity.

This procedure resulted in eight coarse-grained classes: Person (PER), Organisation (ORG), Location (LOC), Geo-Political (GPE), Facility (FAC), Vehicle (VEH), Weapon (WEA), Product (PRO) and Not-Named-Entity (NOT).

2.2 Annotation Strategy and Evaluation[1]

Two Arabic native speakers were involved in the annotation process, using the modified NEs taxonomy in Section 2.1. It was decided that a reasonable goal would be to annotate 4,000 documents and the annotators used a self-developed annotation tool to facilitate the annotation process and both annotators were given guidelines, which clearly defined the distinguishing features of each class, including a practical method to pursue the annotation.

[1] The annotated dataset of Arabic Wikipedia articles is freely available at http://www.cs.bham.ac.uk/~fsa081/

The annotators were initially given the first 500 articles to annotate as a training session, in order to evaluate and identify limitations that might then be expected to manifest during the annotation process. It was expected that there would be a lower level of agreement between them in this round. In order to evaluate the inter-annotator agreement between the annotators we used the Kappa Statistic (Carletta, 1996).

The overall annotation task, including the training session, was divided into three cycles to ensure the resolution of any difficulties the annotators might encounter. After each cycle, the Kappa was calculated and reported. Table 1 summarises the results when evaluating the inter-annotator agreement for each coarse-grained level.

Class	Kappa n = 500	Kappa n = 2000	Kappa n = 4000
PER	98	99	99
ORG	76	94	97
LOC	76	92	97
GPE	97	99	99
FAC	54	88	96
VEH	100	100	100
WEA	85	85	99
PRO	91	97	98
NOT	91	98	98

TABLE 1 - Inter-annotator agreement in coarse-grained level.

The percentage of the coverage of the articles referring to named entities in the annotated documents is 74%.

2.3 Features Representation

The features representation affected the way the classification process was modelled in order to classify given Wikipedia articles and to then produce the mapped named entity class for this article; otherwise the article would not relate to a named entity. In this research, we conducted a comprehensive investigation to evaluate different methods of representing features in order to evaluate those most suitable to our task.

- **Term Presence (TP):** For each given document, the feature representation was simply counted by examining the presence of the tokens in the document. There was no consideration given regarding the frequency of the tokens.
- **Term Frequency (TF):** This represents how many times the tokens in our corpus were found in a given document.

  For a given set of documents $D = \{d_1, d_2, \ldots, d_n\}$ where n is the number of documents. The term frequency (TF) for a given token (t) is calculated thus

$$TF(t, D) = \sum_{d \in D} frequency(d, t)$$

- **Term Frequency-Inverse Document Frequency (TF-IDF):** This reveals how important a given token is to a document within the corpus. It involves scaling down the most frequent words across the documents while scaling up rare ones. The (TF-IDF) is then calculated by multiplying the (TF) with the inverse document frequency (IDF) as follows:

$$TF - IDF(t) = TF(t,d) \times IDF(t)$$

where:

$$idf(t) = log \frac{|D|}{1 + |\{d: t \in d\}|}$$

where $|\{d: t \in d\}|$ is the number of documents the term (t) appears in.

2.4 Features Engineering

The nature of AW articles differs compared with traditional newswire documents, as newswire articles have a tendency to be of a particular length and size due to certain externally imposed conditions. This does not apply to AW, and so some articles are very short while others are very long. Therefore, this necessitates a careful extraction of the most useful textual elements of offer a good representation of the article. Moreover, being able to minimise the size of the dataset, while maintaining representation of semantic knowledge can also accelerate the classification running time.

We believe that using complete tokens in articles contributed surplus noisy data to the model. Therefore, we manually investigated several AW articles of different types in order to define appropriate locations. We decided to compile our raw dataset based on four different locations, based on specific aspects of the AW articles. These are the articles title (t), the first sentence (f), category links (c) and infobox parameters (p).

Although the dataset was modelled as a bag-of-words, we were interested in investigating the optimum features set used within this representation, so as to yield the highest performance for our classification model. The feature sets presented below either involve eliminating or augmenting data, i.e. features, which have been defined as either language-dependent or independent:

- **Simple Features (SF):** This represents the raw dataset as a simple bag of words without further processing. The idea in this case is to evaluate the nature of the full word representation of the AW articles in this task.
- **Filtered Features (FF):** In this version, the following heuristic has been applied in order to obtain a filtered version of the dataset:
    1. Removing the punctuation and symbols (none alphabetical tokens).
    2. Filtering stop words.
    3. Normalising digits where each number has been converted into a letter (d). If we have a date such as 1995, this will be normalised to “dddd”.
- **Language-dependent Features (LF):** Both Syiam et al. (2006) and El-Halees (2007) report the usefulness of the stem representation of the token, in reference to news-wire corpora. This value would not apply to AW. Therefore, we aimed to investigate the effect of applying shallow morphological processing. We relied on the NLTK::ISRIStemmer package (Bird et al., 2009) which is based on the algorithm proposed by Taghva et al. (2005).
- **Enhanced Language-dependent Features (ELF):** This features set was processed in several steps, which are explained below:

1. Tokenising all tokens within the data set using the AMIRA tokeniser developed by Diab (2009), applying the tokenisation scheme of (Conjunction + Preposition + Prefix) instead of stemming. Tokenisation then revealed valuable information such as (Det) and valuable proclitic data, such as the plural noun phrases in AW articles' categories.
2. Using the same tool to assign the part of speech (POS) for each token would allow filtering of the dataset by involving only nouns (for instance) in the classifier.
3. Isolation of tokens based on their locations: this is a novel idea for representing the dataset. The intent in this case being to isolate similar tokens, which appear in different locations on a given document. The intuition behind this is that some tokens that appear in a particular location, i.e. title, first sentence, categories and infobox, of the AW articles, are more discriminative in certain location rather than the whole article. The idea with isolation would be to attach to each token an identifier, i.e. (t) for title, (f) for first sentence, (c) for category and (i) for infobox, to act as a header based on the location in which the token appears. The results of the isolation process are shown in Figure 1.

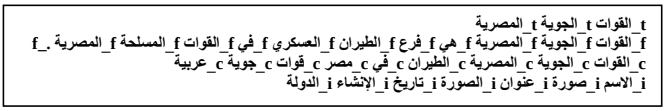

Figure 1: The isolated representation of the article titled "Egyptian Air Force"

In this case example, the feature representation of the token (**المصرية** /AlmSryħ/ 'The Egyptian')[2] presented in the first sentence does not affect, and is not affected by, the same token in the category links or title, even though they have identical glyphs. Surprisingly, the implementation of this idea contributed significant improvements to the classification process.

4. For term presence (TP) only, we applied the most informative features for the top 1000 informative features. To calculate the most informative features we used a Chi Square test (Yang and Pedersen, 1997).

3 Experimentation and Results:

We conducted the experiments by splitting the annotated dataset into training and test sets of 80% and 20% respectively. To the best of our knowledge, there is no similar comparable work for the target language and dataset; therefore we will instead analyse our findings as comprising a comparative study of several properties.

The experiment was designed to evaluate three factors; the features representation, features sets and the probabilistic models. Therefore we extensively use this 3-touple representation to facilitate analysis of the results.

Several text classifiers were applied in order to evaluate performance: Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), and Support Vector Machine (SVM). Since we expected to

[2] Throughout this paper and where appropriate, Arabic words are represented in three variants:
(Arabic word /HSB transliteration scheme (Habash et al., 2007)/ 'English translation')

have a sparse representation of the features, we examined the Stochastic Gradient Descent (SGD) classifier (Bottou, 1991). Moreover, we were not aware of the possibility of applying this classifier to Arabic textual data previously. The experimentation was conducted relying on both Scikit-learn (Pedregosa et al. 2011) and NLTK (Bird et al., 2009).

Since the traditional Naïve Byes classifier relies on term presence we started by evaluating those factors alone. The following table presents the features sets used, in conjunction with three standard metrics, i.e. Precision, Recall and balanced F-measure.

Classifier	Features set	precision	Recall	f1-score
NB	SF	0.60	0.54	0.56
	FF	**0.62**	0.62	0.62
	LF	0.59	0.69	0.63
	ELF	**0.62**	**0.81**	**0.70**

TABLE 2 - The classification results when using Naive Bayes across different features sets where (TP) is applied

Although both FF and ELF have scored identical points, ELF shows significant improvements in the recall and F-measure. This gives the impression that, the enhanced features, i.e. ELF, have boosted the model so as to recall more documents. Table 3 shows the result when applying the remaining classifiers in the case of the TF as the feature representing the backbone.

Features	MNB			SGD			SVM		
set	P	R	F	P	R	F	P	R	F
SF	0.82	0.82	0.81	0.81	0.79	0.77	0.86	**0.87**	0.86
FF	0.82	0.82	0.82	0.87	0.87	0.87	**0.87**	0.86	0.86
LF	0.77	0.76	0.76	0.83	0.83	0.83	0.83	0.83	0.83
ELF	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.87**	**0.87**	**0.87**

TABLE 3 - The classification results using MNB, SGD and SVM over different features sets where (TF) is applied

The tuples {TF, ELF, SGD} and {TF, ELF, MNB} achieved the best result of all the metrics. It is also shown that, MNB has been affected by the feature set used, as it performs slightly better than NB, where LF was used. {TF, SF, SVM} has proven to perform very well by merely using a simple features set. An important point to notice is that, using ELF leads to the highest performance across all classifiers. However, relying on stemming only, as with LF illustrates that there are no such improvements when comparing with other features sets, with the exception of SGD. The results of applying TF-IDF for features representation are shown in Table 4.

Features	MNB			SGD			SVM		
set	P	R	F	P	R	F	P	R	F
SF	0.85	0.85	0.85	0.89	0.89	0.89	0.89	**0.89**	**0.89**
FF	0.86	0.86	0.85	**0.9**	**0.9**	**0.9**	**0.9**	**0.89**	**0.89**
LF	0.79	0.78	0.78	0.86	0.86	0.86	0.85	0.85	0.85
ELF	**0.88**	**0.88**	**0.88**	0.89	0.89	0.89	0.89	0.89	**0.89**

TABLE 4 - The classification results when using MNB, SGD and SVM over different features sets where (TF-IDF) is applied

In the main, all classifiers showed improvements; although this was not the case with {TF-IDF, ELF, MNB} despite MNB scoring better compared with reliance on TF for other features sets. The tuple {TF-IDF, FF, SGD} outperforms all other models where this shows the ability for SGD to generalise the optimum model in order to achieve the highest performance. {TF-IDF, FF, SVM} scored 0.9 on precision, while slightly missing one point on both the recall and F-measures.

4 Discussion:

It was proven that carefully selecting the 3-tuple i.e. {Feature representation, Features set, Statistical model}, yields significant benefits in the sense of overall performance. This can be achieved, in this study, by empirically evaluating the effects of each tuple. Otherwise, closely inspecting the dataset is mandatory but this seems unfeasible in most practical applications.

We have demonstrated that it is possible to achieve a high level performance by compiling parts of the raw dataset as explained in Section 2.4; it is therefore beneficial in minimising the running time of the whole classification process. We doubt, however, if similar heuristics would be valid over a news-wire based corpus.

Due to the nature of AW, it is evident that TP is not the right choice for feature representation. To understand this point, see Figure 1 where the words (**القوات** /AlqwAt/ 'The Troop') and (**قوات** /qwAt/ 'Troop') have been repeated four and two times respectively. Meanwhile, (TF) and (TF-IDF) representation have exploited the redundancy of tokens and showed dramatic improvements of all features and sets.

Language-dependent features have the tendency to cause different affects. Shallow morphological analysis of tokens, i.e. stemming, show no further improvements across features representation and classifiers. Unlike stemming, tokenisation and filtering the analysis POS of the type "Nouns" is superior.

5 Related Work

An early contribution to Arabic NER was made by Maloney and Niv (1998). This involved a combination of a morphological analyser and a pattern recognition engine, the former being responsible for identifying the start and the end of a token, and the latter for identifying the corresponding pattern applied.

Abuleil (2004) developed an NE tagger for QA systems. The aim of this being to eventually acquire a database of names by utilising keywords and specific verbs to identify potential NE. Once this was achieved a directed graph could then be used to delineate the relationship between words contextualised in phrases. Finally, the verification step is accomplished by applying rules to the names.

Shaalan and Raza (2007) compiled a large lexicon list dedicated to personal names forming a gazetteer, extracted from different resources. The gazetteer contained over 472000 entries, including first, middle and last names, job titles and country names. They applied a regular expression rule to identify the availability of personal names in the selected context. Given that Arabic is a highly inflectional language and has relatively free word ordering, designing generic hand-crafted rules is challenging. Traboulsi (2009) partially utilised contextual clues to identify

personal names, by identifying reporting verbs as keywords preceding a personal name. Building a reasonably large gazetteer requires, in addition to time and effort, various additional resources to assure a wide coverage of entities. Elsebai et al. (2009) took a different approach; merging parts of speech with manually created keywords and heuristic rules, without using a gazetteer.

A slightly wider granular NER was later proposed by Shaalan and Raza (2009), with the ability to identify ten different types of named entities. This extended the work of Shaalan and Raza (2007), which relied on gazetteers and lists of rules derived from large resources. A disambiguation method was used to resolve the inevitability of lexical overlap.

Four different machine learning methods have been utilised: Maximum Entropy (Benajiba and Rosso, 2007), Structured Perceptrons (Farber et al., 2008), Support Vector Machines (Benajiba et al., 2008) and Conditional Random Fields (AbdelRahman et al., 2010). It is difficult to judge which approach is the most effective, as the results are inevitably affected by the set of features used. Thus, researchers tend to empirically test different sets of features using various approaches, aiming to achieve an optimum result, for instance as in the work of Benajiba et al. (2008).

In terms of detecting named entities and delimiting their boundaries in Arabic Wikipedia, the work presented by Attia et al. (2010) relies on multilingual interlinks by utilising capitalisation as well as a specific set of heuristics. Recently, Mohit et al. (2012) developed a semi-supervised approach to detect named entities in the Arabic Wikipedia. A self-training algorithm combined with cost function was presented to solve the issue regarding low recall when training on out of domain data. Alotaibi and Lee (2012) presented an approach to identify the NEs in AW. The idea is centred on transforming the news-wire domain for binary NEs detection. A CRF sequence model has been used in order to perform the classification.

Dakka and Cucerzan (2008) presented the first work in which Wikipedia was exploited for a NE task. Their goal was to classify Wikipedia articles into traditional NE semantic classes. For this purpose a set of 800 random articles was manually annotated in order for use with the classifier. Naïve Bayes and the Support Vector Machine (SVM) were chosen as the statistical interface exploiting a specific set of features; such as bag-of-words, structured data, unigram and bigram context. Recently, Saleh et al. (2010) proposed a similar approach to classifying multilingual Wikipedia articles into traditional NE classes. The assumption in that case was that most Wikipedia articles relate to a named entity. Therefore, sets of structured and unstructured data have been extracted so as to be used as a features set when using a support vector machine. Among these features are bag-of-words, category links and infobox attributes. Thus multilingual links are exploited in order to map classified articles for different languages.

6 Conclusion

In the study detailed in this paper we tackled the problem of mapping Arabic Wikipedia articles into a predefined set of NEs classes. We modelled this problem as a document classification issue and comprehensive experiments were empirically conducted in order to evaluate several properties concerning the classification task. Despite our prior assumptions, the use of enhanced language-dependent features did not always lead the best performance especially when combined with the TDF-IDF statistic. More generally we showed that automatic named entity classification can be done on the Arabic Wikipedia with reasonable accuracy.

References

AbdelRahman, S., Elarnaoty, M., Magdy, M., and Fahmy, A. (2010). Integrated machine learning techniques for Arabic named entity recognition. IJCSI International Journal of Computer Science, 7(4):27–36.

Abuleil, S. (2004). Extracting names from Arabic text for question-answering systems. In Proceedings of Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval (RIAO 2004), pages 638–647, Avignon, France.

ACE (2008). Ace (automatic content extraction) English annotation guidelines for entities. [accessed 12 June 2012].

Alotaibi, F. and Lee, M. (2012). Using Wikipedia as a resource for Arabic named entity recognition. pages 27–34, Rabat, Morocco. In Proceeding of the 4th International Conference on Arabic Language Processing (CITALA12).

Attia, M., Toral, A., Tounsi, L., Monachini, M., and van Genabith, J. (2010). An automatically built named entity lexicon for Arabic. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10), Valletta, Malta. European Language Resources Association (ELRA).

Benajiba, Y., Diab, M., and Rosso, P. (2008). Arabic named entity recognition: An svm-based approach. In Proceedings of 2008 Arab International Conference on Information Technology (ACIT), pages 16–18, Amman, Jordan. Association of Arab Universities.

Benajiba, Y. and Rosso, P. (2007). Anersys 2.0: Conquering the NER task for the Arabic language by combining the maximum entropy with POS-tag information. In Proceedings of the Workshop on Natural Language-Independent Engineering, 3rd Indian Int. Conf. on Artificial Intelligence, IICAI-2007, Pune, India.

Bird, S., Klein, E., and Loper, E. (2009). Natural language processing with Python. O'Reilly Media.

Bottou, L. (1991). Stochastic gradient learning in neural networks. Proceedings of Neuro-*Nımes*, 91.

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. Computational linguistics, 22(2):249–254.

Dakka, W. and Cucerzan, S. (2008). Augmenting Wikipedia with named entity tags. In

Proceedings of the 3rd International Joint Conference on Natural Language Processing, pages 545–552, Hyderabad, India. Asian Federation of Natural Language Processing.

Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In 2nd International Conference on Arabic Language Resources and Tools.

El-Halees, A. (2007). Arabic text classification using maximum entropy. The Islamic University Journal (Series of Natural Studies and Engineering) Vol, 15:157–167.

Elsebai, A., Meziane, F., and Belkredim, F. Z. (2009). A rule based persons names Arabic extraction system. Communications of the IBIMA, 11(6):53–59.

Farber, B., Freitag, D., Habash, N., and Rambow, O. (2008). Improving NER in Arabic using a morphological tagger. In Proceedings of the Sixth International Language Resources and *Evaluation (LREC'08)*, pages 2509–2514, Marrakech, Morocco. European Language Resources Association (ELRA).

Habash, N., Soudi, A., and Buckwalter, T. (2007) On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, Arabic Computational Morphology: Knowledge-based and Empirical Methods. Springer.

Maloney, J. and Niv, M. (1998). Tagarab, a fast, accurate Arabic name recognizer using high-precision morphological analysis. In Proceedings of the Workshop on Computational Approaches to Semitic Languages, pages 8–15, Montreal, Quebec, Canada. Association for Computational Linguistics.

Mohit, B., Schneider, N., Bhowmick, R., Oflazer, K., and Smith, N. (2012). Recall-oriented learning of named entities in Arabic Wikipedia. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), pages 162–173. Citeseer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12:2825-2830.

Saleh, I., Darwish, K., and Fahmy, A. (2010). Classifying Wikipedia articles into NE's using SVM's with threshold adjustment. In Proceedings of the 2010 Named Entities Workshop, pages 85–92, Uppsala, Sweden. Association for Computational Linguistics.

Shaalan, K. and Raza, H. (2007). Person name entity recognition for Arabic. In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, pages 17–24, Prague, Czech Republic. Association for Computational Linguistics.

Shaalan, K. and Raza, H. (2009). NERA: Named entity recognition for Arabic. Journal of the American Society for Information Science and Technology, 60:1652–1663.

Syiam, M., Fayed, Z., and Habib, M. (2006). An intelligent system for Arabic text categorization. International Journal of Intelligent Computing and Information Sciences, 6(1):1–19.

Taghva, K., Elkhoury, R., and Coombs, J. (2005). Arabic stemming without a root dictionary. In Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, volume 1, pages 152–157. IEEE.

Traboulsi, H. (2009). Arabic named entity extraction: A local grammar-based approach. In Proceedings of the 2009 International Multiconference on Computer Science and Information Technology (IMCSIT 2009), pages 139–143, Mragowo, Polland.

Yang, Y. and Pedersen, J. (1997). A comparative study on feature selection in text categorization. In Machine Learning-International Workshop Then Conference, pages 412–420. Morgan Kaufmann Publishers, INC.