EXPLORING THE EFFECTS OF STEMMING ON ARABIC NAMED ENTITY RECOGNITION

Ismail El bazi and Nabil Laachfoubi

Univ Hassan 1, IR2M Laboratory, 26000 Settat, Morocco

ABSTRACT

Stemming is the process of reducing words to their stems or roots. Due to the morphological richness and complexity of the Arabic language, stemming is an essential part of most Natural Language Processing (NLP) tasks for this language. In this paper, we study the impact of different stemming approaches on the Named Entity Recognition (NER) task for Arabic and explore the merits, limitations and differences between light stemming and root-extraction methods. Our experiments are evaluated on the standard ANERCorp dataset as well as the AQMAR Arabic Wikipedia Named Entity Corpus.

KEYWORDS
Natural Language Processing, Named Entity Recognition, Stemming, Arabic

1. INTRODUCTION

The Named Entity Recognition task aims to identify and categorize proper nouns and important nouns in a text into a set of predefined categories of interest such as persons, organizations, locations, etc. NER is an important preprocessing step in many NLP applications, including Information Retrieval[1], Machine Translation[2], Summarization [3] or Question Answering[4].

The majority of the work on NER focuses primarily on English language. Over the last decade, Arabic NER has started to gain significant momentum and a lot of work has been done for this language with the increased availability of annotated corpora. Arabic is a Semitic language with a complex morphology and a highly inflectional nature[5]. The concatenative morphology in Arabic makes it possible for words to get formed by attaching affixes to the root. These characteristics cause data sparseness and therefore require very large corpus for training Arabic NER systems in comparison with English NER systems. To overcome this obstacle for Arabic language, one proposed solution is performing stemming.

In this paper, we investigate the impact of various stemming approaches on Arabic NER. These approaches include light stemming methods (Light1, Light2, Light3, Light8, Light10 and Motaz) and root-extraction methods (KHOJA, ISRI and Tashaphyne).

Our main goal is to measure the difference between the light stemmers and root-extraction stemmers and check which one is more suitable for the Arabic NER task.

The remainder of the paper is organized as follows: Section 2 gives background about Arabic Language and the challenges related to Arabic Named Entity Recognition. Section 3 surveys previous work on Arabic NER. Section 4 presents the different stemmers used in this study. In Section 5 the experimental setup is described, and in Section 6 the experimental results are reported. Section 7 provides final conclusions.

DOI : 10.5121/ijaia.2016.7104

2. BACKGROUND

2.1. The Arabic Language

The Arabic language is a Semitic language spoken in the Arab World, a region of 22 countries with a collective population of 300 million people. It is ranked the fifth most used language in the world and one of the six official languages of the United Nations[6]. Arabic is written from right to left using the Arabic script. It has 28 letters, 25 are constants and 3 are long vowels.

With regards to language usage, there are three forms of the Arabic language:

- **Classical Arabic (CA)**: is the formal version of the language. It has been in usage in the Arabian Peninsula for over 1500 years. CA is fully vowelized and most Arabic religious texts are written in this form;

- **Modern Standard Arabic (MSA)**: is the primary written language of the media and education as well as the major medium of communication for public speaking and broadcasting in all Arab countries. MSA is the common language of all the Arabic speakers and the most widely used form of the Arabic language. The main differences between CA and MSA are basically in style and vocabulary, but in terms of linguistic structure, MSA and CA are quite similar[5].This is the form studied in this paper;

- **Dialectal Arabic (DA)**: is the day to day spoken form of the language used in the informal communication. It is not taught in schools or standardized. DA is region-specific that differs not only from one area of the Arab world to another, but also across regions in the same country. This creates a state of diglossia [7] where the MSA is the shared written language among all Arabs, but it is not a native language of anyone.

2.2. Challenges in Arabic Named Entity Recognition

The NER task is considerably more challenging when it is targeting a morphologically rich language such as Arabic for four main reasons:

- **Absence of Capitalization**: Unlike Latin script languages, Arabic does not capitalize proper nouns. Since the use of capitalization is a helpful indicator for named entities[8], the lack of this characteristic increases the complexity of the Arabic NER task;

- **Agglutination**: The agglutinative nature of Arabic makes it possible for a Named Entity (NE) to be concatenated to different clitics. A preprocessing step of morphological analysis needs to be performed in order to recognize and categorize such entities. This peculiarity renders the Arabic NER task more challenging;

- **Optional Short Vowels**: Short vowels (diacritics) are optional in Arabic. Currently, most MSA written texts do not include diacritics, this causes a high degree of ambiguity since the same undiacritized word may refer to different words or meanings. This ambiguity can be resolved using contextual information[9];

- **Inherent Ambiguity in Named Entities**: Proper nouns can also represent regular words. For example, the word "راشد" which means "adult" can be a person name or an adjective. Also, Arabic can face the problem of ambiguity between two or more NEs. In the example “تيمور” (Timur), it is both a person name and a location name which create a conflict situation for the Arabic NER task;

- **Spelling Variants**: In Arabic, as for many other languages, an NE can have multiple transliterations. The lack of standardization leads to many spelling variants of the same word with the same meaning. For example, the transliteration of the Person name ’Samuel’ may produce these spelling variants:
“صموئيل”, ”صامويل ”, ”ساموبل ”, “سمول” or “صمول”.

3. RELATED WORK

Significant amount of work has been done in the last decade for Arabic NER task. The first attempt to handle Arabic NER was TAGARAB system[10]. It was a rule-based system and achieved 85% F-measure on a corpus of 3,214 tokens of the AI-Hayat newspaper. Mesfar [11] presented a rule-based NER system for Arabic using a combination of NooJ syntactic grammars and a morphological analysis . In [12], Shaalan and Raza introduced a system called NERA using a rule-based approach. It is divided into three components: gazetteers, local handcrafted grammars, and a filtering mechanism. NERA obtained 85.58% F-measure on a manually constructed corpus.

In addition to rule-based approach, numerous research studies have been conducted for Arabic NER using Statistical Learning (SL). Benajiba et al. [13] developed an Arabic NER system (ANERsys 1.0) based on n-grams and Maximum Entropy (ME). The system can classify four types of NEs: Person, Location, Organization and Miscellaneous. The authors also introduced a new corpus (ANERcorp) and gazetteers (ANERgazet). In order of overcome some issues in detecting long NEs, Benajiba et al. [14] proposed a new version of their system (ANERsys 2.0), which use two-steps mechanism for NER and exploit the POS feature to enhance the NE boundary detection. Benajiba and Rosso [15] changed the probabilistic model from ME to Conditional Random Fields (CRF) in an attempt to improve the accuracy of ANERsys. The feature set used include POS tags, Base Phrase Chunking (BPC), gazetteers, and nationality information. The CRF-based system achieved an overall 79.21% F-measure on ANERCorp corpus. In [16], Abdul-Hamid and Darwish suggested a simplified feature set that attempt to overcome some of the orthographic and morphological complexities of Arabic without the use of any external lexical resources. The proposed set of features included the leading and trailing character n-grams in words, word unigram probability and the word length feature.

A hybrid approach combining both Statistical Learning and Rule-based has been also used for Arabic NER. Abdallah et al. [17] presented a hybrid NER system for Arabic. The SL-based component uses Decision Tree, while the rule-based component is a re-implementation of the NERA system [12] using the GATE framework. Recently, Shaalan and Oudah [18] published a hybrid system that produces state-of-the-art results with an overall 90.66% F-measure on ANERCorp dataset.

Stemming and lemmatization was already incorporated in Arabic NER systems. Abdul-Hamid and Darwish [16] used a reimplementation of the stemmer proposed by Lee et al. [19] in their CRF-based system . Al-Jumaily et al. [20] created a real time NER system for Arabic text mining and adapted the Khoja stemmer [21] for the stemming step. In [22], a light stemmer [23] was used to produce stem feature for the evaluation of the newly created Wikipedia-derived corpus (WDC). Zirikly and Diab [24] presented a NER system for Dialectal Arabic using lemmas generated from MADAMIRA tool[25].

4. STEMMERS

Various stemmers were developed for Arabic. They can be grouped in two types; the first type is light stemmers which remove affixes (i.e. prefixes and suffixes) from the words, while the second type are called root-extraction stemmers (i.e. heavy stemmers) which extract the root of the words.

In this section, we briefly describe the different stemmers used in this paper.

4.1. KHOJA Stemmer

Khoja stemmer [21] is one of the early and most powerful stemmer developed for Arabic[26],[27. Khoja begins by removing diacritics, punctuation, non-characters and the longest suffix and prefix of the input word, and then attempts to extract the root by matching the remaining word with the verbal and noun predefined patterns. Finally, the extracted root gets validated against a list of correct Arabic roots. If no root is found, then the word is left intact. This stemmer relies on several linguistic resources such as a list of all punctuation characters, diacritic characters, definite articles, and 168 stop words.

4.2. ISRI Stemmer

ISRI stemmer [28] is a root-extraction stemmer. ISRI shares many characteristics with Khoja stemmer[21]. However, the main difference is that ISRI does not linguistically validate the extracted roots against any type of dictionaries. It starts by removing diacritics, normalizing Hamza to one form (أ) and removing prefixes of length three and length two prefixes in that order. Then it removes the connector (و) if it precedes a word beginning with (و) and normalize all the forms of Hamza to (ا). Finally, ISRI searches for possible matches within a group of patterns, if there is no match; it successively attempts to trim single-character affixes and reiterate the search. The stemming process should be stopped when it either matches a pattern and extracts the relevant root, or when the remaining length of the word is three or less characters.

4.3. Tashaphyne Stemmer

Tashaphyne [29] is an Arabic Light Stemmer. It uses two lists of prefixes and suffixes to detect the affixes attached to a given word and find the root. In addition to root extraction, Tashaphyne can be used for light stemming as well.

4.4. Motaz Stemmer

Motaz stemmer [30] provides both root extraction and light stemming. The root extraction part is an implementation of Khoja stemmer [21] with the only difference is using another stopwords list. For the light stemming part, it is an implementation of the Light10 Arabic light stemming algorithm proposed by Larkey and colleagues in [31]. Before applying the Light10 algorithm, Motaz stemmer normalize the input word by removing diacritics, replacing all the forms of Hamza with (ا), replacing (ة) with (ه) and replacing (ى) with (ي).

4.4. Larkey's Light Stemmers

Light1, Light2, Light3, Light8 and Light10 are a set of light stemmers created by Larkey and colleagues [31] for Arabic Information Retrieval. They all follow the same steps as described in [31]:

- Remove و ("and") for light2, light3, and light8, and light10 if the remainder of the word is three or more characters long.
- Remove any of the definite articles if this leaves two or more characters.
- Go through the list of suffixes once in the (right to left) order indicated in Table 1, removing any that are found at the end of the word, if this leaves two or more characters.

Table 1. Strings removed by Larkey's light stemming [31]

	Remove prefixes	Remove Suffixes
Light1	ال، وال، بال، كا ل، فا ل	none
Light2	ال، وال، بال، كا ل، فا ل ، و	none
Light3	"	ة ، ه
Light8	"	ها، ان، ات ، ون ، ين، ية ، يه ، ة ، ه ، ي
Light10	ال،وال، بال، كا ل، فا ل، لل ، و	"

5. EXPERIMENTAL SETUP

5.1 NER System

Our NER system is based on Conditional Random Fields sequence labeling as described in[32]. CRF is considered by many authors as one of the most competitive algorithms for NER [6],[33]. We use the following feature set for our experiments:

- **Word** : The surrounding words of a context window -1,…,+1 ;
- **Stem**: The surrounding stems of a context window -1,…, +1. The stemming approaches used are described in section 4;
- **Affixes**: Prefixes and suffixes of the stem were used. Their length ranges from 1 to 4;
- **Character n-grams** : The leading and trailing bigrams, trigrams and 4-grams characters as reported in [16].

5.2. Corpora

In this paper we use two datasets: ANERCorp and AQMAR Arabic Wikipedia Named Entity Corpus (AQMAR).

ANERcorp is a news-wire domain corpus of more than 150,000 words annotated especially for the NER task by Benajiba and colleagues [13]. It is a commonly used corpus in the literature for comparing with existing systems and it became a standard dataset for the Arabic NER task.

Table 2. Number of different NEs in ANERcorp [16]

Named Entity	Number
Persons	689
Organizations	342
Locations	878

AQMAR Arabic Wikipedia Named Entity Corpus is a 74,000-token corpus of 28 Arabic Wikipedia articles hand-annotated for named entities by Mohit and colleagues [34].
For training and testing, we used a 70/30 split of each dataset.

Table 3. Number of different NEs in AQMAR

Named Entity	Number
Persons	636
Organizations	133
Locations	538

5.3. Tools

In this work, we used the following tools:

- CRF++[1], a CRF sequence labeling toolkit used with default parameters.
- AraNLP [35], a Java-based Library for the Processing of Arabic Text. This library includes a sentence detector, tokenizer, light stemmer, root stemmer, POS tagger, word segmenter, normalizer, and a punctuation and diacritic remover.
- SAFAR [36] , an integrated platform that brings together all layers of Arabic NLP. This platform, includes, a normalizer, sentence splitter, tokenizer, stemmers, syntactic parsers and morphological analyzers.

5.4. Evaluation Metrics

We adopted the strict CoNLL evaluation metric to evaluate our results. This strict metric considers the tagged entity as correct only if it is an exact match of the corresponding entity in the gold data [37] . It is based on the commonly known precision, recall and F-measure which are defined as follows:

$$Recall = \frac{number\ of\ correctly\ recognized\ NEs}{Total\ number\ of\ NEs}$$

$$Precision = \frac{number\ of\ correctly\ recognized\ NEs}{Total\ number\ of\ NEs\ retrieved\ \ by\ the\ system}$$

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

[1] https://code.google.com/p/crfpp/

6. EXPERIMENTS & RESULTS

We adopt a straightforward design for our experiments. In the first experiment, we train a NER model on the training set using each stemming approach. Then, we evaluate these models on the test set. In the second experiment, we combine the stemming approach with the best results in the first experiment with each of the remaining approaches and we train a new NER model on the training set. Then we evaluate these models again on the test set. For all experiments, we use the feature set described in section 5. It's a simplified feature set and should fulfil well the requirements of all our experiments.

The results of our first experiment are shown in Tables 4-5 and Figures 1-2. We can see that even the simplest methods improve the results on both datasets compared to the word-based baseline. The methods based on the light stemming approaches significantly outperform the methods based on root-extraction techniques.

The best results on ANERCorp dataset were achieved using the Light1 stemmer. For AQMAR dataset, the Light1 was edged out slightly by Light2 stemmer.

Generally, the simpler the method is, the better the result is achieved in our tests.

The results of our second experiment are shown in Tables 6-7. We can see that all the stemmer combinations improve the results on both datasets compared to the Light1 stemmer (baseline). The best results on ANERCorp dataset were achieved using the combination of Light1 and Tashaphyne stemmer. For AQMAR dataset, the best results were achieved by combining Light1 with Light8.

Generally, stemmer combinations achieve better results compared to using single stemmer in our tests.

Overall, according to the results of all our experiments, including stems as feature improve the performance of Arabic NER systems specially using a simple approach (aka light stemming). Also, combining different stemming approaches seems to enhance even more the performance of Arabic NER systems.

Table 4. Results for the ANERCorp

	Precision	**Recall**	**F-measure**
Baseline	85.80	40.11	54.18
ISRI	78.29	55.79	65.11
Khoja	77.15	54.14	63.56
Motaz	79.38	56.76	66.12
Tashaphyne	76.80	52.17	62.12
Light1	82.76	59.41	69.10
Light2	81.34	59.15	68.42
Light3	81.18	58.77	68.09
Light8	79.34	56.71	66.07
Light10	79.37	56.78	66.13

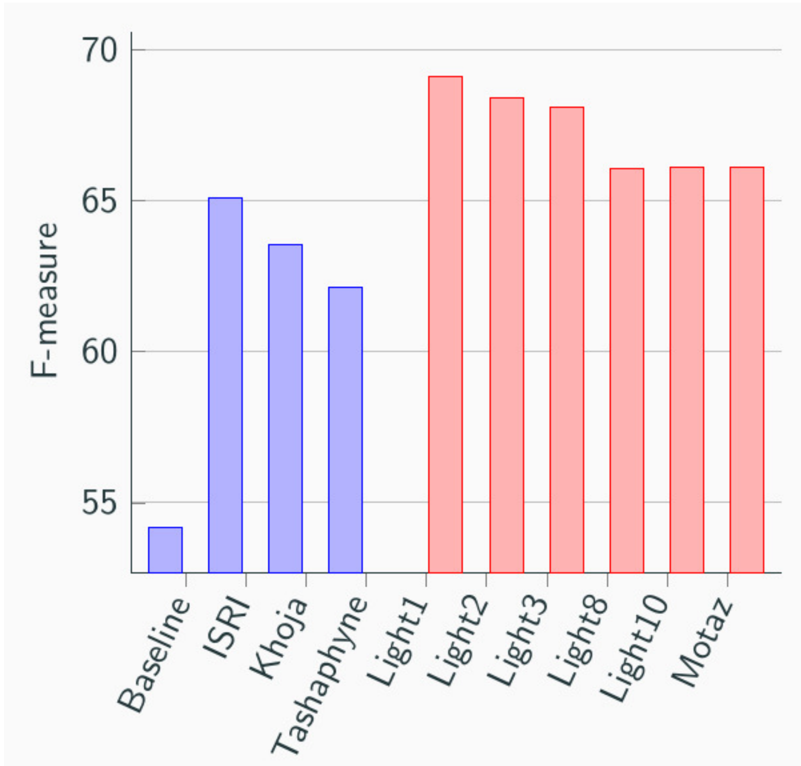

Figure 1. Performance comparison (ANERcorp)

Table 5. Results for the AQMAR

	Precision	**Recall**	**F-measure**
Baseline	74.46	24.15	35.72
ISRI	64.69	41.49	50.39
Khoja	65.27	41.67	50.64
Motaz	70.69	44.29	54.24
Tashaphyne	63.95	38.51	47.67
Light1	72.91	46.73	56.90
Light2	72.45	47.09	57.03
Light3	72.01	46.14	56.15
Light8	70.72	43.86	53.92
Light10	70.79	44.29	54.26

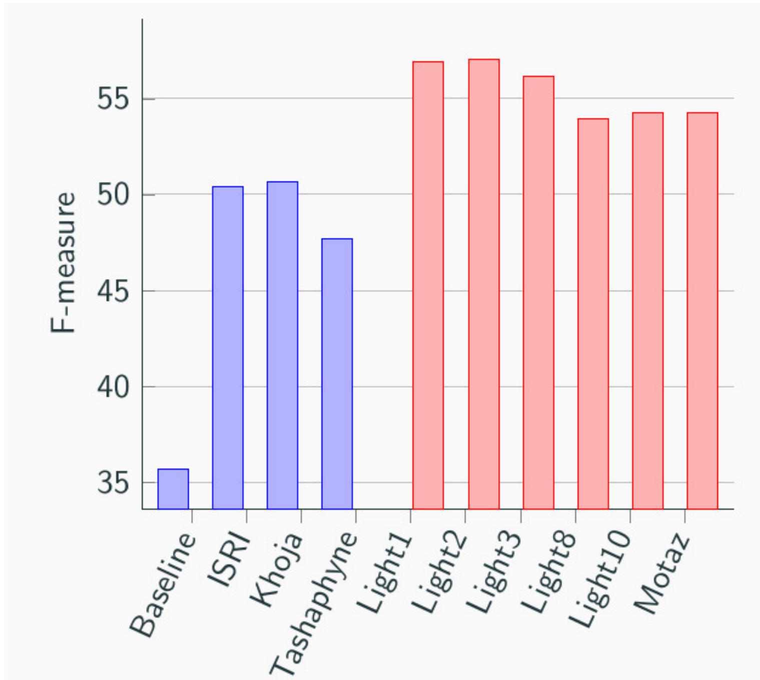

Figure 2. Performance comparison (AQMAR)

Table 6. Stemmer combination results for the ANERCorp

	Precision	Recall	F-measure
Light1 (Baseline)	82.76	59.41	69.10
Light1 + Light2	82.08	60.82	69.80
Light1 + Light3	81.65	60.91	69.71
Light1 + Light8	81.75	61.21	69.95
Light1 + Light10	81.63	61.30	69.96
Light1 + Motaz	81.24	61.14	69.71
Light1 + Khoja	81.03	61.97	70.17
Light1 + ISRI	81.28	60.87	69.55
Light1 + Tashaphyne	81.89	61.82	70.40

Table 7. Stemmer combination results for the AQMAR

	Precision	Recall	F-measure
Light1 (Baseline)	72.91	46.73	56.90
Light1 + Light2	72.74	48.18	57.94
Light1 + Light3	72.68	48.36	58.07
Light1 + Light8	73.31	48.58	58.39
Light1 + Light10	72.98	48.29	58.09
Light1 + Motaz	72.91	48.41	58.16
Light1 + Khoja	71.95	48.96	58.20
Light1 + ISRI	73.12	46.75	56.95
Light1 + Tashaphyne	73.12	46.87	57.07

7. CONCLUSION

We have tested nine different stemming approaches in the Arabic NER task on two datasets ANERCorp and AQMAR. They include Light stemmers and root-extraction stemmers.

The results show that light stemming approaches significantly outperform the root-extraction approaches. All stemming approaches were better than the word-based baseline. The best results were achieved using the Light1 stemmer with 69.10% F-measure on ANERCorp. For AQMAR corpus, the best results were achieved using the Light2 stemmer with 57.03% F-measure. Also, combining different stemming approaches enhance the overall performance of Arabic NER systems.

REFERENCES

[1] H.-H. Chen, Y.-W. Ding, and S.-C. Tsai, “Named entity extraction for information retrieval,” Computer Processing of Oriental Languages, vol. 12, no. 1, pp. 75–85, 1998.
[2] B. Babych and A. Hartley, “Improving machine translation quality with automatic named entity recognition,” in Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT, 2003, pp. 1–8.
[3] C. Nobata, S. Sekine, H. Isahara, and R. Grishman, “Summarization System Integrated with Named Entity Tagging and IE pattern Discovery.,” in LREC, 2002.
[4] D. Mollá, M. Van Zaanen, D. Smith, and others, “Named entity recognition for question answering,” 2006.
[5] K. C. Ryding, A reference grammar of modern standard Arabic. Cambridge University Press, 2005.
[6] I. El bazi and N. Laachfoubi, “RENA: A Named Entity Recognition System for Arabic,” in Text, Speech, and Dialogue, vol. 9302, P. Král and V. Matoušek, Eds. Springer International Publishing, 2015, pp. 396–404.
[7] C. A. Ferguson, “Diglossia,” 1959.
[8] Y. Benajiba, M. Diab, and P. Rosso, “Arabic named entity recognition using optimized feature sets,” in In Proc. of EMNLP’08, 2008, pp. 284–293.
[9] Y. Benajiba, M. Diab, and P. Rosso, “Arabic Named Entity Recognition: A Feature-Driven Study,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. 17, no. 5, pp. 926–934, Jul. 2009.
[10] J. Maloney and M. Niv, “TAGARAB: a fast, accurate Arabic name recognizer using high-precision morphological analysis,” in Proceedings of the Workshop on Computational Approaches to Semitic Languages, 1998, pp. 8–15.
[11] S. Mesfar, “Named Entity Recognition for Arabic Using Syntactic Grammars,” in Natural Language Processing and Information Systems, vol. 4592, Z. Kedad, N. Lammari, E. Métais, F. Meziane, and Y. Rezgui, Eds. Springer Berlin Heidelberg, 2007, pp. 305–316.
[12] K. Shaalan and H. Raza, “NERA: Named Entity Recognition for Arabic,” Journal of the American Society for Information Science and Technology, vol. 60, no. 8, pp. 1652–1663, 2009.
[13] Y. Benajiba, P. Rosso, and J. BenedíRuiz, “ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy,” in Computational Linguistics and Intelligent Text Processing, vol. 4394, A. Gelbukh, Ed. Springer Berlin Heidelberg, 2007, pp. 143–153.
[14] 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.,” in IICAI, 2007, pp. 1814–1823.
[15] Y. Benajiba and P. Rosso, “Arabic named entity recognition using conditional random fields,” in Proc. of Workshop on HLT & NLP within the Arabic World, LREC, 2008, vol. 8, pp. 143–153.
[16] A. Abdul-Hamid and K. Darwish, “Simplified Feature Set for Arabic Named Entity Recognition,” in Proceedings of the 2010 Named Entities Workshop, 2010, pp. 110–115.
[17] S. Abdallah, K. Shaalan, and M. Shoaib, “Integrating Rule-Based System with Classification for Arabic Named Entity Recognition,” in Computational Linguistics and Intelligent Text Processing, vol. 7181, A. Gelbukh, Ed. Springer Berlin Heidelberg, 2012, pp. 311–322.

[18] K. Shaalan and M. Oudah, “A hybrid approach to Arabic named entity recognition,” Journal of Information Science, vol. 40, no. 1, pp. 67–87, 2014.

[19] Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan, “Language Model Based Arabic Word Segmentation,” in Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, 2003, pp. 399–406.

[20] H. Al-Jumaily, P. Martínez, J. Martínez-Fernández, and E. Van der Goot, “A real time Named Entity Recognition system for Arabic text mining,” Language Resources and Evaluation, vol. 46, no. 4, pp. 543–563, 2012.

[21] S. Khoja and R. Garside, “Stemming arabic text,” Lancaster, UK, Computing Department, Lancaster University, 1999.

[22] M. Althobaiti, U. Kruschwitz, and M. Poesio, “Automatic Creation of Arabic Named Entity Annotated Corpus Using Wikipedia,” in Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), 2014, pp. 106–115.

[23] L. S. Larkey, L. Ballesteros, and M. E. Connell, “Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis,” in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002, pp. 275–282.

[24] A. Zirikly and M. Diab, “Named Entity Recognition for Dialectal Arabic,” ANLP 2014, p. 78, 2014.

[25] A. Pasha, M. Al-Badrashiny, M. Diab, A. E. Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, “MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic,” in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14), 2014.

[26] I. A. Al-Sughaiyer and I. A. Al-Kharashi, “Arabic morphological analysis techniques: A comprehensive survey,” Journal of the American Society for Information Science and Technology, vol. 55, no. 3, pp. 189–213, 2004.

[27] L. S. Larkey and M. E. Connell, “Arabic information retrieval at UMass in TREC-10,” DTIC Document, 2006.

[28]K. Taghva, R. Elkhoury, and J. Coombs, “Arabic stemming without a root dictionary,” in Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, 2005, vol. 1, pp. 152–157 Vol. 1.

[29] T. Zerrouki, “Tashaphyne, Arabic light Stemmer/segment.” 2010.

[30] M. K. Saad and W. Ashour, “Arabic morphological tools for text mining,” Corpora, vol. 18, p. 19, 2010.

[31] L. Larkey, L. Ballesteros, and M. Connell, “Light Stemming for Arabic Information Retrieval,” in Arabic Computational Morphology, vol. 38, A. Soudi, A. den Bosch, and G. Neumann, Eds. Springer Netherlands, 2007, pp. 221–243.

[32] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” in Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 282–289.

[33] D. Lin and X. Wu, “Phrase clustering for discriminative learning,” in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, 2009, pp. 1030–1038.

[34] B. Mohit, N. Schneider, R. Bhowmick, K. Oflazer, and N. A. Smith, “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, 2012, pp. 162–173.

[35] M. Althobaiti, U. Kruschwitz, and M. Poesio, “AraNLP: a Java-Based Library for the Processing of Arabic Text,” in Proceedings of the 9th Language Resources and Evaluation Conference (LREC), 2014.

[36] Y. Souteh and K. Bouzoubaa, “SAFAR platform and its morphological layer,” in Eleventh Conference on Language Engineering ESOLEC’2011, 2011.

[37] E. F. Tjong Kim Sang, “Introduction to the CoNLL-2002 Shared Task: Language-independent Named Entity Recognition,” in Proceedings of the 6th Conference on Natural Language Learning - Volume 20, 2002, pp. 1–4.