Voting Strategies for Arabic Named Entity Recognition using Annotation Schemes

Ikram Belhajem
Faculty of Sciences, Mohammed V University, Rabat, Morocco
i.belhajem@um5r.ac.ma (corresponding author)

Received: 6 August 2024 | Revised: 31 August 2024 | Accepted: 14 September 2024

 DOI: https://doi.org/10.48084/etasr.8645

ABSTRACT

Named Entity Recognition (NER) seeks to identify and classify NEs into predefined categories and is an important subtask in information extraction. Many annotation schemes have been proposed to assign suitable labels for multiword NEs within a given text. This study proposes a method to combine the results of different annotation schemes (IOB, IOE, IOBE, IOBS, IOES, and IOBES) for Arabic NER (ANER). Three voting strategies are explored, namely, majority voting, weighted voting, and weighted voting-based Particle Swarm Optimization (PSO), applied to Conditional Random Fields (CRF) classifiers, each corresponding to a certain annotation scheme. The experimental results showed that majority voting can be considered an effective combination strategy to enhance the performance of ANER systems.

Keywords-information extraction; named entity recognition; machine learning; conditional random fields; annotation schemes, voting strategies

I. INTRODUCTION

The concept of Named Entity (NE), introduced in [1], involves the recognition of person names, organization names, geographic location names, time, currency, and percentage expressions within structured or unstructured text using SGML markup. Since then, NE Recognition (NER) has emerged as one of the most important subtasks in information extraction that seeks to identify and classify all NEs in a document into predefined categories (e.g. person, location, organization, and miscellaneous) [2]. NER is considered a crucial preprocessing phase in many Natural Language Processing (NLP) applications to enhance their overall performance, such as Information Retrieval (IR) [3], Machine Translation (MT) [4], Question Answering (QA) [5], and Search Results Clustering (SRC) [6]. Many studies have been conducted on NER for many different languages, including Arabic. Arabic is a Semitic language, spoken by more than 360 million people in more than 30 countries around the world [7], and is a highly inflected language with a rich morphology and complex syntax. Some specificities of Arabic make it a highly challenging language to deal with in the context of NER [8], such as lack of capitalization, agglutination, optional short vowels or diacritics, lack of uniformity in writing styles, and ambiguity between NE types In the literature, three major approaches are used to perform Arabic NER (ANER):

- The rule-based approach [9, 10] relies on manually crafted local grammatical rules written by expert linguists. Thus, any adjustment required for rule-based NER systems is labor-intensive and time-consuming [8].
- The Machine Learning (ML) based approach [11, 12] depends on different learning algorithms that use feature sets obtained from annotated texts with NEs to build statistical models for NER systems. Therefore, solid linguistic knowledge is not required to develop ML-based NER systems that are adaptable and easily maintained with insignificant effort and minimum time provided sufficient large tagged datasets.
- Hybrid approaches [13] combine the previous two approaches by providing the rule-based output as a feature used by the ML classifier. Such an integration aims to overcome the limitations of each approach when processed individually and to improve the overall performance of NER systems.

The need for a large amount of annotated data is a prerequisite for training and testing NER models. Furthermore, since many NEs consist of multiple words, it is not feasible to annotate subsequent entities with the same type [14]. To this end, several annotation schemes have been developed to label multiword NEs in an attempt to increase recognition performance. The primitive simplest annotation scheme applied in NER is the IO format [15], in which I represents each word within the entity and O stands for non-entity words. However, this scheme is unable to correctly represent multiword NEs, as it cannot recognize subsequent entities of the same type. In [2], the IOB scheme was adopted to annotate the corpus, and since then it has been the most widely used format in NER systems. In this scheme, each word in the text is assigned to a certain tag, be it the Beginning (B), the I, or the O of the NE [16]. However, some studies explored the combination of results from different annotation schemes and analyzed their impact on the performance of NER systems for multiple languages.

In [17], the nearest-neighbor memory-based learner was presented as a basic classifier to perform language-independent NER. This learner was applied to five annotation schemes, namely, IOB1, IOB2, IOE1, IOE2, and O+C. Its performance was improved using three additional techniques: cascading, feature selection, and system combination with majority voting. This approach was evaluated on Spanish and Dutch datasets, showing that combining the results of the five cascaded systems corresponding to different annotation schemes with majority voting achieved better performance than the individual learners. In [18], two different strategies were used in a NER tagger, namely majority voting and Decision Tree (DT), to combine the results of IOB1, IOB2, IOE1, IOE2, IOBES, and IO. The CoNLL dataset was used to train and test the conditional Markov model, showing that the F1 score of majority voting on various tagging schemes tends to perform close to the best of these schemes, while the DT had poor performance. In [19], a two-stage ensemble approach was applied for clinical NER. In the first stage, the Support Vector Machine (SVM) algorithm was used to learn four base classifiers with different annotation schemes, namely IOB2, IOE2, IOBE, and IOBES. Then, the outputs of these classifiers were combined using majority voting and stacking separately in the second stage. This approach was evaluated on the i2b2 dataset, and the results showed that both ensemble classifiers outperformed each of the base classifiers, while the stacking technique achieved the best F score. In the same perspective, in [20] a new segment representation was proposed to improve multiword biomedical NER, and majority voting was used to combine the outputs of the IOB2, IOBES, and FROBES segment representations. Experiments were carried out using Artificial Neural Networks (ANNs) based on Bidirectional Long Short-Term Memory (Bi-LSTM) and Conditional Random Fields (CRF) on the JNLPBA and i2b2 datasets. The results showed that combining the outputs of different segment representations with majority voting achieved better performance than each baseline model.

Regarding the Arabic language to our knowledge, this is the first work to investigate the impact of using multiple voting strategies to integrate various annotation schemes on the performance of ANER. This study explored the following annotation schemes :

- IOB.
- IOE is similar to the IOB format but replaces the B tag with the E tag to indicate the End of the NE.
- IOBE is a variation of the IOB scheme that additionally distinguishes the last word of multiword NEs with the E tag to have more information concerning the entity boundaries.
- IOBS labels the entities identically to IOB in addition to the Single (S) tag, which is used to identify NEs including only a single word.
- IOES is an extension of the IOE scheme that adds the S tag for single-word NEs.
- IOBES is a further extension to the IOB scheme that consists of five tags, namely B, I, and E for multiword NEs, S for one-word NEs, and O for non-entity words.

The combination of annotation scheme outputs is based on three different voting strategies, namely majority voting, weighted voting [21], and weighted voting-based PSO, for ANER. For other languages, the proposed approach differs from the existing literature by introducing other annotation schemes (i.e., IOBS and IOES) and by exploring further voting strategies, namely weighted voting and weighted voting-based PSO.

II. METHODOLOGY

The proposed system consists of four main phases as illustrated in Figure 1. In the first phase, data annotation is carried out to convert the IOB format applied on the original corpus to the rest of the tagging formats, namely IOE, IOBE, IOBS, IOES, and IOBES. The second phase involves the extraction of NER features related to each word in the text. In the third phase, the extracted features are fed into the CRF classifier. Finally, the outputs of the classifiers corresponding to each annotation scheme are combined with different voting strategies to identify Arabic NEs.

A. Data Annotation

This study used the ANERcorp [22] dataset, which classifies NEs into the four categories defined in the CoNLL-2002, namely Person, Location, Organization, and Miscellaneous. The corpus follows the IOB annotation scheme to assign every word in the text to a specific tag (i.e., B, I, or O). ANERcorp contains the words of the text along with their corresponding label that indicates both the boundary tag along the NE class, and it can be one of the following: B-PERS, I-PERS, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC, or O. Concerning the different annotation schemes adopted, Python scripts were developed to convert the IOB format to IO, IOE, IOBE, IOBS, IOES, and IOBES. This results in several datasets generated from the original ANERcorp, each corresponding to a certain annotation scheme. To illustrate the difference between these formats, Table I shows an example of tagging a text fragment with each annotation scheme. Table II presents the number of annotation labels in each dataset that is calculated given the number of tags per annotation scheme and the number of NE categories.

B. NER Features

The features in NER are characteristic attributes of words designed for algorithmic consumption. The feature vectors are fed to the NER classifier as input data, representing each word to be categorized by one or more Boolean or binary, numerical, and nominal values [8]. The proposed approach used the following features for the NER task.

- Context Words: These are preceding and subsequent words related to the current NE within a context window (the window size was chosen to be ±1). This feature is used under the observation that the surrounding words carry effective information to identify NEs.
- Word Prefix and Suffix: Extracting word prefixes and suffixes, if present, can be a good sign of capturing the presence of NE, as most ANEs have no prefix or suffix [23]. This is generated by Tashaphyne [24], which is an Arabic light stemmer and segmentor.

- Word Length: This is a binary feature used to check if the length of the current word is greater than a predefined threshold (set to 3 characters in this study). This is based on the fact that very short words are rarely NEs.
- Part Of Speech (POS) information: This feature identifies the POS category (e.g., noun, verb, adjective, preposition, etc.) of the current word and its surrounding words (one previous and one after in this approach). The MADAMIRA tool [25] was used to generate POS information, which is a tool for morphological analysis and disambiguation in the Arabic language.
- Morphological Features: This is a set of morphological information generated by MADAMIRA and based on exploiting the rich morphological features of Arabic.
- Corresponding English Capitalization: This is a binary feature that checks the capitalization of the English translation corresponding to the current Arabic word. It is used to compensate for the missing capitalization feature in the Arabic language, based on the English gloss generated by MADAMIRA. If the translated word begins with a capital letter, it is most likely an NE [16].
- Gazetteers: This is a set of binary features that indicate whether the word exists within each of the various predefined lists of typed NEs. This study used ANERGaz [22], which consists of three gazetteers. These gazetteers were used for complete names of people, locations (e.g., names of continents, countries, cities, etc.), and organizations (e.g., names of companies, football teams, and other organizations).
- Contains Digit: This is a binary feature used to examine whether the word contains any digit (0-9), which helps recognize miscellaneous NEs, such as time expressions, measurement expressions, and numerical numbers [26].
- Character n-grams: This is a set of features consisting of current leading and trailing character unigrams, bigrams, trigrams, and quadrigrams. These character n-gram features implicitly capture valuable morphological and orthographic clues that would indicate the presence or absence of NEs [27].
- Stop Words: This binary feature checks whether the word is in the stop words list. Stop words cannot be part of NEs. In this study, the stop words list was collected based on [28] and consists of 1383 words including prepositions, pronouns, conditional pronouns, verbal pronouns, and adverbs.

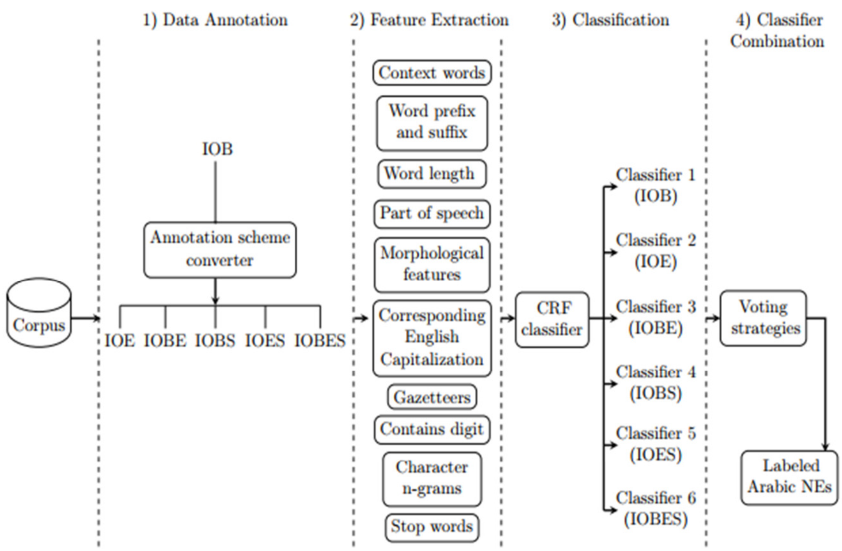

Fig. 1. ANER system architecture.

TABLE I. ANNOTATION OF A FRAGMENT TEXT WITH VARIOUS SCHEMES

Word	IOB	IOE	IOBE	IOBS	IOES	IOBES
عبر	O	O	O	O	O	O
فضائية	O	O	O	O	O	O
الجزيرة	B-ORG	E-ORG	B-ORG	S-ORG	S-ORG	S-ORG
أكد	O	O	O	O	O	O
السيد	O	O	O	O	O	O
حسن	B-PERS	I-PERS	B-PERS	B-PERS	I-PERS	B-PERS
نصر	I-PERS	I-PERS	I-PERS	I-PERS	I-PERS	I-PERS
الله	I-PERS	E-PERS	E-PERS	I-PERS	E-PERS	E-PERS
الأمين	O	O	O	O	O	O
العام	O	O	O	O	O	O

TABLE II. NUMBER OF ANNOTATION LABELS FOR EACH DATASET

Dataset	Number of annotation labels
IOB dataset	9
IOE dataset	9
IOBE dataset	13
IOBS dataset	13
IOES dataset	13
IOBES dataset	17

C. Classification

This approach adopted CRF, which is a supervised ML algorithm, to identify Arabic NEs, since the NER task can be regarded as a sequence labeling problem to assign a specific label to each word in a given input sequence. CRF [29] are discriminative probabilistic models that are well suited for segmenting and labeling sequence data and have been applied successfully for several NLP tasks, in particular NER [30]. They are a type of conditionally trained undirected graphical models whose output nodes represent the label sequence while the input nodes correspond to the data sequence. Therefore, CRF aim to find a y that maximizes the conditional probability $P(y|x)$ of a label sequence $y = y_1, \ldots, y_T$, given an input sequence $x = x_1, \ldots, x_T$ as defined in

$$P(x|y) = \frac{1}{Z(x)} exp(\sum_{t=1}^{T} \sum_{k=1}^{N} \lambda_k f_k(y_{t-1}, y_t, x, t)) \quad (1)$$

where T is the sequence length, N is the number of features, $f_k(y_{t-1}, y_t, x, t)$ is a feature function whose value may range from $-\infty$ to $+\infty$ but it is often binary, λ_k represents a learned weight assigned to each feature function f_k, and $Z(x)$ is a normalization factor expressed as

$$Z(x) = \sum_y exp\left(\sum_{t=1}^{T} \sum_{k=1}^{N} \lambda_k \ f_k(y_{t-1}, y_t, x, t)\right) \quad (2)$$

D. Classifier Combination

Voting is the most obvious way to combine classifiers, which is considered a simple solution that allows each classifier to vote for the class of its choice [31]. In this phase, the outputs of classifiers related to the IOE, IOBE, IOBS, IOES, and IOBES formats are converted back to the IOB format using Python scripts. Then, three different voting strategies are used to combine the results of these classifiers to recognize Arabic NEs:

- Majority voting: Each classifier is given one vote, and the final output label with the highest number of votes is chosen.
- Weighted voting: This strategy assigns various weights to the base classifiers based on specific criteria, and the final output label with the highest weighted vote is selected. In this work, each classifier was weighted according to its micro-averaged F score value obtained in the training phase.
- Weighted voting-based PSO: The weighted voting strategy coupled with PSO was used for weight optimization. Since the strengths of PSO encompass fast convergence, simplicity of implementation, and computational efficiency in terms of both speed and memory [32], it is used to obtain the optimal weights associated with individual classifiers. For this, an initial swarm is generated randomly and the particles are the weights to learn. Each particle i moving around the search space is characterized by a position vector x_i, a velocity vector v_i, and a position at which the best fitness $pbest_i$ is achieved by the particle. Besides, the global best position $gbest_i$ represents the position yielding the lowest error among all $pbest_i$. At each iteration, the particles of the swarm are updated according to the following equations:

$$v_i(k+1) = wv_i(k) + c_1 r_1(k)\big(pbest_i - x_i(k)\big) + c_2 r_2(k)\big(gbest_i - x_i(k)\big) \quad (3)$$

$$x_i(k+1) = x_i(k) + v_i\,(k+1) \quad (4)$$

where c_1 and c_2 are positive constants known as acceleration coefficients, $r_1(k)$ and $r_2(k)$ are uniformly distributed random numbers generated separately in the range $[0, 1]$, and w is the inertia weight employed to control the impact of the previous history of velocities on the current one. The update process is repeated until a maximum number of iterations is reached or an acceptable $gbest$ is achieved.

III. EXPERIMENTS AND RESULTS

Experimental tests were carried out on the ANERcorp dataset [22] using CRF to examine the impact of applying various voting strategies to combine different annotation schemes on ANER performance. The ANERcorp is a freely available annotated corpus that is manually collected from different article types extracted from various newspapers to obtain a more generalized corpus. It contains more than 150,000 tokens of which 11% are NEs distributed as 39% for person, 30.4% for location, 20.6% for organization, and 10% for miscellaneous. It is composed of a training corpus and a test corpus with 125,000 and 25,000 tokens each, respectively.

In all experiments, the metrics used to evaluate the ANER system performance were the micro-averaged precision, recall, and F score. Precision refers to the percentage of NEs identified by the evaluated system that are correct, recall is the percentage of NEs present in the corpus that are found by the system [2], while the F score is defined as a harmonic mean between precision and recall with equal weight.

During the training phase, different extracted features were used as input to the CRF classifier, whose implementation was carried out using the Python sklearn-crfsuite sequence classification library [33]. The L-BFGS algorithm [34] was used to train the CRF model with the maximum number of iterations set to 100 and allowing for all possible transition features, while the values for the L1 and L2 regularization coefficients were set to 0.1. The CRF models for the six annotation schemes were learned based on the ANERCorp training corpus using 10-fold cross-validation (splits of the original data into 10 folds of approximately equal size - at each iteration, one fold is considered as the test set, while the remaining nine folds are used as the training set). Subsequently, the results of these models were combined with different voting strategies to recognize Arabic NEs. For weighted voting, Table III presents the weights assigned to each base classifier, which are their micro-averaged F-score values obtained in the training phase. Table IV shows the implementation parameters for weighted voting-based PSO. Table V summarizes the results of voting strategies over various CRF classifiers, each corresponding to a specific annotation scheme. It should be noted that the O tag was excluded when calculating the evaluation metrics to avoid distorting results, as the vast majority of ANERcorp tokens are tagged as O. The results show that the voting strategies produced better results than the individual classifiers in terms of precision, recall, and F score. Furthermore, majority voting outperforms weighted voting by up to 0.12% precision and 0.06% F score, with equal recall, while weighted voting-based PSO gives results close to weighted voting, with a simple decrease of 0.01% in precision, 0.02% in recall, and 0.01% in F score.

In the prediction phase, the trained CRF models were applied on the test ANERcorp corpus to classify unseen Arabic NEs, and their outputs were combined using different voting strategies. Table IV presents the precision, recall, and F score results of the voting strategies. The majority voting performed close to the best individual classifier corresponding to the IOB annotation scheme with a decrease of 0.96% in precision, 0.37% in F score, and equal recall. Weighted voting outperformed weighted voting-based PSO by up to 0.26% in precision, 0.06% in recall, and 0.13% in F score.

TABLE III. IMPLEMENTATION PARAMETERS OF WEIGHTED VOTING STRATEGY

Classifiers	Weights
Classifier 1 (IOB)	0.7632
Classifier 2 (IOE)	0.7623
Classifier 3 (IOBE)	0.7554
Classifier 4 (IOBS)	0.7538
Classifier 5 (IOES)	0.7512
Classifier 6 (IOBES)	0.7459

TABLE IV. IMPLEMENTATION PARAMETERS OF WEIGHTED VOTING-BASED PSO

Parameter	Value
Swarm size	20
Inertia weight w	0.729
Positive constant c_1	1.49445
Positive constant c_2	1.49445
Numbers r_1 and r_2	Random
Number of iterations	100

TABLE V. RESULTS OF VOTING STRATEGIES OBTAINED ON THE ANERCORP TRAINING CORPUS

		Precision (%)	Recall (%)	F score (%)
Individual classifiers	Classifier 1 (IOB)	82.79	70.79	76.32
	Classifier 2 (IOE)	82.80	70.63	76.23
	Classifier 3 (IOBE)	82.33	69.78	75.54
	Classifier 4 (IOBS)	82.09	69.68	75.38
	Classifier 5 (IOES)	81.75	69.47	75.12
	Classifier 6 (IOBES)	81.31	68.89	74.59
Voting strategies	Majority voting	99.92	99.98	99.95
	Weighted voting	99.80	99.98	99.89
	Weighted voting based PSO	99.79	99.96	99.88

TABLE VI. RESULTS OF VOTING STRATEGIES OBTAINED ON THE ANERCORP TESTING CORPUS

		Precision (%)	Recall (%)	F-measure (%)
Individual classifiers	Classifier 1 (IOB)	80.24	63.06	70.62
	Classifier 2 (IOE)	78.68	62.46	69.64
	Classifier 3 (IOBE)	78.66	61.39	68.96
	Classifier 4 (IOBS)	77.26	60.62	67.93
	Classifier 5 (IOES)	76.55	60.68	67.70
	Classifier 6 (IOBES)	74.52	59.76	66.33
Voting strategies	Majority voting	79.28	63.06	70.25
	Weighted voting	79.69	62.43	70.01
	Weighted voting-based PSO	79.43	62.37	69.88

Given these results on the ANERcorp training and testing sets, majority voting can be considered an important combination strategy over CRF base classifiers to enhance the performance of the proposed ANER system. Another finding is that majority voting achieves better performance than weighted voting and weighted voting-based PSO. These conclusions are close to those presented in some previous studies, which highlighted the high performance of majority voting compared to individual classifiers [17, 19, 20] or a certain voting strategy [18]. Furthermore, using micro-averaged F score values to weight individual classifiers leads to better results than using the optimal weights found with PSO.

IV. CONCLUSION

This study applied various voting strategies, namely majority voting, weighted voting, and weighted voting-based PSO, to combine the results of CRF classifiers corresponding to six different annotation schemes (IOB, IOE, IOBE, IOBS, IOES, and IOBES) for ANER. The experimental tests carried out on the ANERcorp training and testing sets show that adopting majority voting as a combination strategy over CRF base classifiers can considerably enhance the performance of the proposed ANER system. In addition, majority voting achieved better results compared to other combination strategies, namely weighted voting and weighted voting-based PSO. Future directions involve investigating the selection and combination of other heterogeneous classifiers to analyze their impact on this ANER system.

REFERENCES

[1] R. Grishman and B. Sundheim, "Message Understanding Conference-6: a brief history," in *Proceedings of the 16th Conference on Computational Linguistics*, Copenhagen, Denmark, 1996, vol. 1, Art. no. 466, https://doi.org/10.3115/992628.992709.

[2] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, Edmonton, Canada, 2003, vol. 4, pp. 142–147, https://doi.org/10.3115/1119176.1119195.

[3] A. Ababneh, J. Lu, and Q. Xu, "Arabic Information Retrieval: A Relevancy Assessment Survey," *Proceedings of the International Conference on Information Systems Development (ISD)*, Katowice, Poland, Sep. 2016.

[4] A. Alqudsi, N. Omar, and K. Shaker, "Arabic machine translation: a survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 549–572, Dec. 2014, https://doi.org/10.1007/s10462-012-9351-1.

[5] L. Abouenour, K. Bouzoubaa, and P. Rosso, "IDRAAQ: New Arabic Question Answering system based on Query Expansion and Passage Retrieval," in *CLEF 2012 Evaluation Labs and Workshop*, Rome, Italy, 2012.

[6] Y. Benajiba, M. Diab, and P. Rosso, "Arabic Named Entity Recognition: A Feature-Driven Study," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 926–934, Jul. 2009, https://doi.org/10.1109/TASL.2009.2019927.

[7] "Arabic - Worldwide distribution," *Worlddata.info*. https://www.worlddata.info/languages/arabic.php.

[8] K. Shaalan, "A Survey of Arabic Named Entity Recognition and Classification," *Computational Linguistics*, vol. 40, no. 2, pp. 469–510, Jun. 2014, https://doi.org/10.1162/COLI_a_00178.

[9] W. Zaghouani, "RENAR: A Rule-Based Arabic Named Entity Recognition System," *ACM Transactions on Asian Language Information Processing*, vol. 11, no. 1, pp. 1–13, Mar. 2012, https://doi.org/10.1145/2090176.2090178.

[10] M. Oudah and K. Shaalan, "NERA 2.0: Improving coverage and performance of rule-based named entity recognition for Arabic," *Natural Language Engineering*, vol. 23, no. 3, pp. 441–472, May 2017, https://doi.org/10.1017/S1351324916000097.

[11] Y. Benajiba and P. Rosso, "Arabic Named Entity Recognition using Conditional Random Fields," in *Proceedings of the Workshop on HLT and NLP within the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, pp. 143–153.

[12] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, Jan. 2022, https://doi.org/10.1109/TKDE.2020.2981314.

[13] M. Oudah and K. Shaalan, "A Pipeline Arabic Named Entity Recognition using a Hybrid Approach," in *Proceedings of the Coling Organizing Committee*, Mumbai, India, 2012, pp. 2159–2176.

[14] M. Konkol and M. Konopík, "Segment Representations in Named Entity Recognition," in *Text, Speech, and Dialogue*, Pilsen,Czech Republic, 2015, pp. 61–70, https://doi.org/10.1007/978-3-319-24033-6_7.

[15] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proceedings of the fifth conference on Applied natural language processing*, Washington, DC, USA, 1997, pp. 194–201, https://doi.org/10.3115/974557.974586.

[16] L. A. Ramshaw and M. P. Marcus, "Text Chunking Using Transformation-Based Learning," in *Natural Language Processing Using Very Large Corpora*, S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky, Eds. Dordrecht, Netherlands: Springer Netherlands, 1999, pp. 157–176.

[17] E. F. Tjong Kim Sang, "Memory-based named entity recognition," in *proceeding of the 6th conference on Natural language learning - COLING-02*, Stroudsburg, PA, USA, 2002, vol. 20, pp. 1–4, https://doi.org/10.3115/1118853.1118878.

[18] V. Krishnan and V. Ganapathy, "Named entity recognition," Stanford University, 2005.

[19] H. Nayel and H. L. Shashirekha, "Improving NER for clinical texts by ensemble approach using segment representations," in *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, 2017, pp. 197–204.

[20] H. A. Nayel, H. L. Shashirekha, H. Shindo, and Y. Matsumoto, "Improving Multi-Word Entity Recognition for Biomedical Texts," in *Proceedings International Journal of Pure and Applied Mathematics*, 2018.

[21] E. Jamalian and R. Foukerdi, "A Hybrid Data Mining Method for Customer Churn Prediction," *Engineering, Technology & Applied Science Research*, vol. 8, no. 3, pp. 2991–2997, Jun. 2018, https://doi.org/10.48084/etasr.2108.

[22] Y. Benajiba, P. Rosso, and J. M. BenedíRuiz, "ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy," in *Computational Linguistics and Intelligent Text Processing*, Mexico City, Mexico, 2007, pp. 143–153, https://doi.org/10.1007/978-3-540-70939-8_13.

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

[24] T. Zerrouki, "Tashaphyne, Arabic light stemmer." 2012, [Online]. Available: https://pypi.python.org/pypi/Tashaphyne/0.2.

[25] A. Pasha *et al.*, "MADAMIRA: 9th International Conference on Language Resources and Evaluation, LREC 2014," in *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, 2014, Art. no. 1094–1101.

[26] A. Ekbal and S. Bandyopadhyay, "Named Entity Recognition using Support Vector Machine: A Language Independent Approach," *International Journal of Electrical, Computer, and Systems Engineering*, vol. 4, no. 2, 2010.

[27] A. Abdul-Hamid and K. Darwish, "Simplified feature set for Arabic named entity recognition," in *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden, Apr. 2010, pp. 110–115.

[28] I. A. El-Khair, "Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study," *International Journal of Computing & Information Sciences*, vol. 4, no. 3, pp. 119–133, 2006.

[29] 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*, Mar. 2001, pp. 282–289.

[30] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, Edmonton, Canada, 2003, vol. 4, pp. 188–191, https://doi.org/10.3115/1119176.1119206.

[31] M. Althobaiti, U. Kruschwitz, and M. Poesio, "Combining Minimally-supervised Methods for Arabic Named Entity Recognition," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 243–255, May 2015, https://doi.org/10.1162/tacl_a_00136.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, Australia, 1995, vol. 4, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[33] "sklearn-crfsuite — sklearn-crfsuite 0.3 documentation." https://sklearn-crfsuite.readthedocs.io/en/latest/.

[34] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, Aug. 1989, https://doi.org/10.1007/BF01589116.