International Journal of Robotics and Control Systems
Vol. 5, No. 2, 2025, pp. 937-952
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

An Integrated Deep Learning Framework Combining LSTM-CRF, GRU-CRF, and CNN-CRF with Word Embedding Techniques for Arabic Named Entity Recognition

Mahdi Ahmed Ali [a,1,*], Ahmed Bahaaulddin A. Alwahhab [b,2], Yagoub Farjami [c,3]

[a] Technical College of Engineering, Middle Technical University, Baghdad, Iraq
[b] Technical College of Management, Middle Technical University, Baghdad, Iraq
[b] Department of computer and IT, University of Qom, Qom, Iran
[1] maaah@mtu.edu.iq; [2] ahmedbahaaulddin@mtu.edu.iq; [3] farjami@qom.ac.ir
* Corresponding Author

ARTICLE INFO

Article history
Received December 27, 2024
Revised February 25, 2025
Accepted March 22, 2025

Keywords
Arabic NLP;
Conditional Random Field;
Deep Learning (DL);
Named Entity Recognition;
Word Embedding

ABSTRACT

Named entity recognition (NER) is the main function of natural language processing (NLP) and has many applications. Arabic NER systems aim to identify and classify Arabic NEs in Arabic text, which provide unique problems due to the language's complex morphology and syntactic structures. This paper provides an integrated deep learning system that incorporates three deep learning architectures—LSTM-CRF, GRU-CRF, and CNN-CRF—as well as three word embedding techniques: GloVe, Word2Vec, and FastText, all trained on Arabic corpus. To develop NER state-of-the-art in Arabic language, the present paper proposed a 3-stage process of pre-processing, feature extraction, and a combination of various deep network schemes. In the preprocessing section, operations such as removing irrelevant words, correcting words, etc. will be used to improve the system's efficiency. In the feature extraction section, three-word embedding methods, Glove, word2vec, and fasttext, which are trained with Arabic texts, are used, and finally, three LSTM-CRF, GRU-CRF, and CNN-CRF models are trained with each word embedding, and the results they are combined. Experimental results on benchmark dataset, ANERcorp show that our methodology is effective, with an accuracy of 94.39%, which outperforms other cutting-edge methods. However, combining multiple deep learning models with word embeddings increases computational complexity and resource requirements, potentially complicating implementation in resource-constrained contexts. Future efforts will concentrate on optimizing the framework to lower computational costs while keeping good performance.

1. Introduction

NEs refer to the textual references through accurate names, including the first and last names, companies, and locations. Diagnosing NEs in unstructured text and grouping them in predefined name groups is called Named Entity Recognition (NER) [1], [2]. NER is an important subtask of Natural Language Processing that identifies and categorizes named entities in unstructured text. While much research has been done on NER in languages such as English, Arabic NER presents distinct issues due to the language's rich morphology, orthographic ambiguity, and lack of capitalization to

distinguish proper nouns. Firstly, task of NER was defined at the Sixth Message Understanding Conference (MUC-6). However, the text could include one/more names' kinds like Organization, Person, Sports, Location also a lot of other names from special fields. Such kinds of names are known as Named Entities (NE). NER looks for automatically recognizing and grouping such names in text into predefined levels. There has been a significant process in Arabic NER over the last ten years also the presented systems have accepted different techniques of NEs that could be hardly grouped into the methods based on the rule, the strategies of Machine Learning (ML). ML strategies are more beneficial since the system could be trained and simply developed for different fields of language [3], [4].

Despite advances in NER using typical ML methodologies, these methods frequently fail to handle Arabic's complex language aspects. Deep learning (DL) models, particularly those that incorporate advanced word embedding techniques, have demonstrated promising success in overcoming these constraints. However, there is a considerable research gap in successfully merging different deep learning architectures with various word embedding approaches to improve Arabic NER performance. Prior to the widespread usage of DL approaches, early NER research concentrated on increasing manual extraction techniques. As the popularity of DL has grown in recent years, the use of NER functions has grown significantly. Based on DL evolution, such techniques can be classified as neural language model-based, embedding-based, multitask learning, attention mechanism-based, word, CNN-based, RNN, sequence-to-sequence-based, pre-trained language models, and prompt-based methods. In addition, there is a collection of GAN-based algorithms that solve a number of issues with NER data production. Sequence annotation, sequence-to-sequence, span-based, hypergraph and translation-based approaches, segment graph, and translation-based strategies are the categories into which such model-based methods could be categorized. There are a few papers on flat entity recognition that use significant models in the late NER development step presented DL [5], [6].

A variety of issues arise as NER systems are enlarged. One example is the confusion surrounding NE kinds. For instance, Sydney can be a company, city in Australia, a female's name. distinguishing among the same entities' kinds could be hard, coping with these ambiguities makes context info necessary also subject matter expertise. Also, diagnosing accurate NE limits in a text could be hard, particularly while coping with entities which include a lot of words/ apply non-standard spellings. Observing NEs where an entity is nested in the other one, like "Alan Smith, Hilton Corp. CFO " and out-of-vocabulary entities which do not show in data of training, current additional problems. For improving successful and resilient systems of NER, this is important to properly assess the way of addressing such syntactic and semantic problems through combining annotated training data, domain-specified info, skill of language [7], [8].

Word embedding can be done using two fundamental methods: context-dependent (contextualized) and context-independent (classic). In traditional embeddings, a word's representation is distinguished by being distinct for every term and not taking any emerging terms into account. Such outcome in words being recognized with no term contemplation causes decreased accuracy: this is given the model of language and the associated public corpora of text. Classic embedding model samples contain GloVe [5], Word2vec [6], and FastText [7].

GloVe is the only matrix factorization methods' source for "word-context matrix" [9]. Normally, corpus might be scanned as: for each term, identify terms of the context in an environment denoted by the window size before and after the term, meaning that words farther away from the core term would be given less weight. Such attributes are beneficial to recognize attributes of language generally analyzing word frequency across corpora.

Word2vec generates and predicts semantic word contexts using two-layer neural network training [10], [11]. Skip-Gram (SG) predicts a context while the word is provided, while continuing bag-of-words (CBOW) offers a strategy for predicting a word based on context. FastText: Such model shows Skip-Gram Word2vec version; although, instead of processing a word as a whole, each word is treated as being made up of n-grams [12].

This study seeks to close this gap by offering an integrated framework that combines the strengths of LSTM-CRF, GRU-CRF, and CNN-CRF models with word embeddings provided by GloVe, Word2Vec, and FastText. Each of these embedding methods has a distinct advantage: GloVe collects global statistical information, Word2Vec excels at collecting semantic similarities, and FastText efficiently handles out-of-vocabulary words using subword information. Furthermore, the motivation for choosing these deep learning architectures is their complimentary capabilities. LSTM and GRU networks excel at modeling sequential data and capturing long-term dependencies, but CNNs excel at recognizing local patterns and features. The incorporation of Conditional Random Fields (CRF) improves sequence labeling performance by taking into account the dependencies between output labels.

The window size in this instance is at the class of character, similar to the word n-grams. Conversely, Word2vec acts at the word class, while FastText operates at the character class. Such a model learns and recognizes the scheme word sub-words in addition to the entire n-character word order. It applies sub-word info to embed, and ensure that hardly ever applied vocabs could yet be properly predicted. Raised morphological language understanding, with each other with developed provided word tense representations, makes system able to govern unfamiliar vocabs. FastText refers to an approach offered for addressing embedding hardness that hardly ever applied vocabs which may sometimes be poorly assumed [13], [14].

Here, we apply techniques of Glove, word2vec, and fasttext for embedding of word level. In the presented technique, for applying Glove, word2vec, and FastText word embedding methods benefits simultaneously, such 3 techniques' word vectors are applied that are trained with Arabic texts. A thorough literature assessment demonstrates that existing Arabic NER systems either rely largely on single-model architectures or lack the ability to generalize across varied datasets. Unlike earlier research, our approach includes not just several models and embeddings, but also an elaborate preprocessing pipeline to address Arabic's unique linguistic problems, such as morphological normalization, diacritic removal, and context-aware tokenization.

The manuscript is outlined as follows: Section 2 defines the work associated with such a study. We mentioned the issue description and presented a neural network scheme in Section 3. Section 4 presents the study method. Section 5 provides an assessment of the outcome, Section 6 concludes the study.

2. Related Work

Arabic is a hard language that has complicated morphological and orthographic manners, which may make the functions of NER hard. Despite this, today, a large study number exists considering Arabic NER. A summary of the most cited studies exists below beginning from early traditional machine learning Arabic NER techniques' steps with covering the most recent deep learning techniques' works.

Maha Al-Rabiah and Noura Al-Saa Antoun and Wissam [15], [16] described BERT-BGRU's strategy. Outcomes illustrate that provided scheme beat the most developed models of ANER, with F-measure values of 92.28% and 90.68% on the ANERCorp dataset integrated with datasets of ANERCorp and AQMA, respectively.

Shadi Al-Halawi et Chadi Helwe and Ghassan Dib al. [17], [18] presented the method of semi-supervised learning to train scheme of NER with labeled and semi-labeled sets of data as well as BERT. F-measure amounts of 65.5% and 78.6% are obtained from the AQMAR and NEWS datasets, respectively. "Mahdhaoui et al and Abdelkarim Mars. [19], [20] present the dynamic method to learn ANER applying pre-trained language model AraGPT2.

Nayel et al and Zitao Zheng. [21], [22] concentrate on NER for Arabic medical texts, particularly aiming at illness entity identification. This research provides the comparative deep learning methods' analysis containing LSTM-CRF, Long Short-Term Memory (LSTM), Bidirectional LSTM

(BiLSTM), which are applied to the set of data containing Arabic medical texts associated with illnesses. The approach removes requirements for big annotated set if data choosing the most informative instances for annotation given the prediction doubt of a model.

Alsaaran, Alrabiah and Hao Wei [23], [24], examined performance of pre-trained BERT model through fine-tuning for NERA, applying various neural networks architectures. The study that was published in particular NER edition illustrates the way of altering pre-trained BERT language model value for languages with complicated morphology and some resources. They checked 6 far model integrations given the BERT also recognized that BERT-BIGRU-CRF beat others. In addition, scheme of BERT-BIGRU-CRF performed better than BERT lonely and BERT-CRF schemes in case of performance.

Shaker et al and Santosh_Kumar-Birthriya [25], [26], defined ANER set of data that is various and includes nine 9 classes called entities. LSTM and GRU models outcomes illustrate that, they act well in tests and validation in training, showing that models could generalize the comprehension and properly detect entities which are contained in sets of validation and test. Texts are various in seven different domains. 2 schemes (LSTM, GRU) present good results, they can identify entities' names with precision of approximately (80%).

Mahdhaoui et al and Taoufiq El Moussaoui [27], [28], examine developments in ANER through dynamic learning techniques' application and large language models usage, including AraBERT. Present paper examines dynamic learning efficiency through selecting informative instances for annotation while leveraging AraBERT power for developing NER performance.

Alsaaran and Alrabiah and Norah Alsaaran [29], [30], examine classic ANER based on DL using different DNN architectures as well as contextual language schemes given the BERT that is trained in general domain of Arabic text. Through fine-tuning the pre-trained BERT language model, they present 2 models given the RNN to group and recognize named things in Classical Arabic. BGRU/BLSTM model was trained applying pre-trained BERT contextual language model representations, the result attributes were modified applying Classical ANER set of data. In addition, they check different provided BERT-BGRU/BLSTM-CRF models architecture.

Al-Smadi et al and Hoanh-Su Le [31], [32], present novel DL strategy for Standard ANER that performed better than the last outcomes. Basic aim of improving new model is presenting developed fine-grained outcomes for the use in NLP fields. Reported technique included applying transfer learning with DNN for creating Pooled-GRU model which was joined with Multilingual Universal Sentence Encoder.

Youssef et al and Zhang, M.; Geng [33], [34], assessed pooled contextual embeddings and bidirectional encoder representations applying Transformers (BERT) model performance for NER in Arabic. The method proposed is an end-to-end deep learning model that combines pre-trained word embeddings, pooled contextual embeddings, and the BERT model. Embeddings are fed into bidirectional long-short-term memory networks via the conditional random domain. The best model was discovered by investigating several forms of contextual and classical embeddings.

Sadallah et al and Abainia, K [35], [36], show ANER, the Named Entity Recognizer for Arabic. This is trained on approximately a huge set of data (500k Tokens). Additionally, they support 50 various levels of entity, contrary to just four entities supported by the other architectures. They expanded their support for the increasingly well-known Arabizi. They made the system accessible to everyone by using it online with an intuitive user interface.

Ali, Tan and M. N. A. Ali [37], [38], show bidirectional encoder-decoder scheme for addressing ANER problem given the present article about DL that the encoder and decoder are bidirectional LSTMs. Furthermore, embeddings in class of vocab and feature are confirmed and integrated applying attention method in class of embedding. In this strategy of attention, our model might dynamically determine info which should to be applied from component in class of vocab/ feature.

Mousa and Genuario [39], [40], provided classification process offering multiple model which includes BiLSTM as well as sequential CNN cascaded with Radial Basis Function (RBF). Provide scheme performance was compared to stand-alone ML models such as: Multilayer Perceptron (MLP), CNN, K-Nearest Neighbors (KNN), Naïve Bayes (NB), BiLSTM, Support Vector Machine (SVM), RBF.

3. Proposed Method

In the proposed method to recognize entity names (NER) in Arabic texts, a three-step process of pre-processing, feature extraction, and a combination of different deep network models is used. In the preprocessing section, operations such as removing irrelevant words, correcting words, etc. will be used to improve the system's efficiency. In the feature extraction section, three-word embedding methods, Glove, word2vec, and fasttext, which are trained with Arabic texts, are used, and finally, three LSTM-CRF, GRU-CRF, and CNN-CRF models are trained with each word embedding, and the results They are combined. The proposed method for entity name recognition (NER) in Arabic texts includes the following steps:

1. Selection of Arabic text dataset for entity name recognition (NER) and its pre-processing.
2. Extracting the features of the input text with three-word embedding methods: Glove, word2vec, and fastest.
3. Construction of different deep neural network models.

- LSTM-CRF neural network model training with Arabic word2vec input embedding layer.
- GRU-CRF neural network model training with Arabic Glove input embedding layer.
- CNN-CRF neural network model training with Arabic Fasttext input embedding layer.

4. Combining the results of the three models using majority voting.

3.1. Selection of Arabic Text Dataset and its Pre-Processing

ANERcorp is a hand-labeled Arabic sample text corpus designed for use in Arabic Name Entity Recognition (NER) systems. This set is divided into two parts: training and testing, and it was labeled by only one individual to ensure uniformity. This collection contains almost 150 thousand tokens, 11 percent of which are named entities. All badges in this collection are labeled with one of the following: person, location, organization, miscellaneous, or other. The identified entities are distributed as follows:

Person: 39 percent

Organization: 20.6 percent

Location: 30.4 percent

Miscellaneous: 10 percent

Such set contains 316 papers initiated from news companies as well as other online resources.

Normalization of Vocabulary:

In Arabic, a word can be written in various formats. To reduce data dispersion and standardize these terms, data normalization was used to convert all of the various word forms into a standard kind. To achieve data homogeneity, multiple types of words are matched to a single model.

Minor Modifications:

To improve the accuracy and consistency of the data, the following improvements were made to the ANERcorp dataset.

- Correcting minor spelling mistakes in tags.

- Converting dots in the middle of the word (·) and full dots (•) to regular dots (.).
- Remove the empty Unicode character (\u200F).
- Adding a sentence separator after a sequence of one or more periods.
- Segmenting the data set in order. Sentences that contain 5.6 initial words are considered for the training part and the rest for the test part.

Arabic-specific challenges in preprocessing:

Arabic presents unique obstacles in the preprocessing process. These challenges include:

Morphological Complexity: Words can undergo a variety of morphological alterations.

Semantic Ambiguity: Words that appear similar in various situations can have distinct meanings. These issues necessitate tailored processing solutions, which in this case involve word normalization and the removal of duplicate information.

Replicability and Changes to the ANERcorp Dataset:

Since the creation of the ANERcorp dataset in 2008 [41], this dataset has been used as a standard reference for researchers in the field of nominalization recognition in Arabic through out the world. However, over time, this dataset has been copied many times between different users, made minor changes to it, and split into different configurations, making it difficult to fairly compare results between different papers and systems.

Some investigators from CAMeL lab in the year 2020 [42] visited Yassin bin Ajibeh, present dataset producer, for discussing on accurate share and obtain the confirmation. Minor modifications to the original data set were also accepted. Bashir al-Hafani from CAMeL laboratory, in cooperation with Nizar Habash, implemented the decisions made in this version. Changes to the original dataset were allowed, including fixing minor spelling errors in tags and converting certain punctuation marks to standard ones.

3.2. Feature Extraction

To recognize entity names in Arabic language (NER), the next step is to choose a suitable representation to extract features from the desired Arabic text document. This stage plays an important role in deep learning. The present step has an essential role in DL. Here, we apply techniques of fasttext, word2vec, and Glove for word-level embedding. Feature extraction aim is converting unorganized as well as noisy textual data in organized and vector types which could be comprehended using ML mechanisms. In word embedding methods, mapping the data to vectors with lower dimensions, improves the ability to learn the network from the data. These vectors are called "embedding". The first and most famous model in this field was made in 2013 by Mikolov and his colleagues known as word2vec [43], [44].

Since then, with the increasing use of deep learning in many instances, this method has been used for feature extraction. One of the problems with word2vec is that it only extracts word vectors and does not work for sentences. For this purpose, in 2014, Miklow and Lee introduced a word2vec-based doc2vec model, which has recently become very popular among natural language processing researchers, and many works have been done using it [45].

The word2vec uses the role of words in a sentence to extract feature vectors, but in 2015 Facebook researchers introduced a method called fastText that uses wordgrams to extract feature vectors, which in many cases outperforms the model The previous ones were like word2vec. Now, many papers were performed in grouping text applying the mentioned technique. One issue with such techniques refers to broad data volume which is required for training models that are hard to gather. For this reason, Stanford University researchers have introduced a model called GLOVE, which uses English text in Twitter data to extract feature vectors. Glav is now widely used in natural processing problems, especially text classification.

Global Vectors for Word Representation (GloVe) is a method for creating vector models of word embeddings, that is the unsupervised algorithm improved as open-resource project at Stanford University [46], [47].

In vectors achieved from the mentioned scheme, words are provided in a meaningful area that distance among vocabs shows semantic similarity among them.

In the proposed method, to be able to use the advantages of Glove, word2vec, and Fasttext word embedding methods at the same time, the word vectors of these three methods that are trained with Arabic texts are used and for each word, a numerical vector of length 300 for Each of these methods is created.

The approaches GloVe, Word2Vec, and FastText all have limitations. GloVe is ineffective in modeling local and long-term dependencies. Word2Vec has an issue with out-of-vocabulary words, which means it performs poorly when encountering unusual words. Although FastText handles out-of-vocabulary terms better, it has a larger computational complexity than Word2Vec and may operate less efficiently in complicated languages such as Arabic. Furthermore, because of their computational complexity, LSTM and GRU models demand more memory and training time, whereas CNN models are less suitable for dealing with long-term dependencies. When combining these methods, GloVe with LSTM or GRU can enhance accuracy since LSTM models manage long-term dependencies more well. However, merging these models may result in greater computational complexity and resource demands, which can be difficult to manage in resource-constrained applications.

3.3. Construction of Different Deep Neural Network Models

In this step, three different neural network models are created, in the first model, which is called W2V-LSTM-CRF, the input layer is the embedding vectors obtained by the Word2vec method, and the next layer is a two-way LSTM neural network, and at the end, a The CRF layer is positioned to predict NER tags. Present scheme framework is illustrated in Fig. 1. CRF (Conditional Random Field) layer refers to DNN's layer kind which is applied for modelling dependencies among network results. Such layer is normally located at the end of network, after layers of feature extraction, and its task is to improve the prediction accuracy of the network by taking into account local and long-range dependencies between tokens [48].

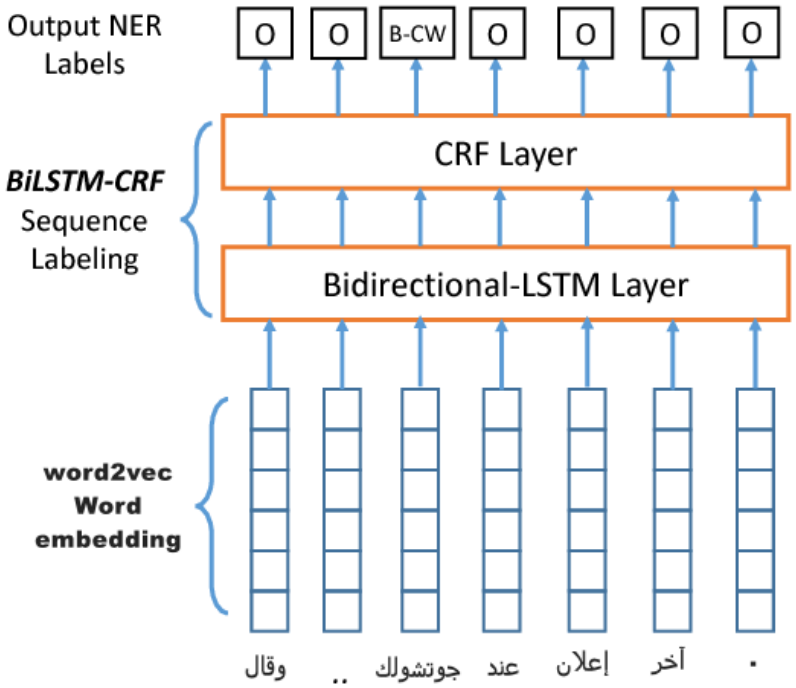

Fig. 1. W2V-LSTM-CRF network architecture in the proposed method

The second model is called GLOVE-GRU-CRF. The input layer is the embedding vectors obtained from the GLOVE method, and the next layer is a two-way GRU neural network, at the end, a CRF layer is placed to predict NER labels. Present model framework is illustrated in Fig. 2 [49].

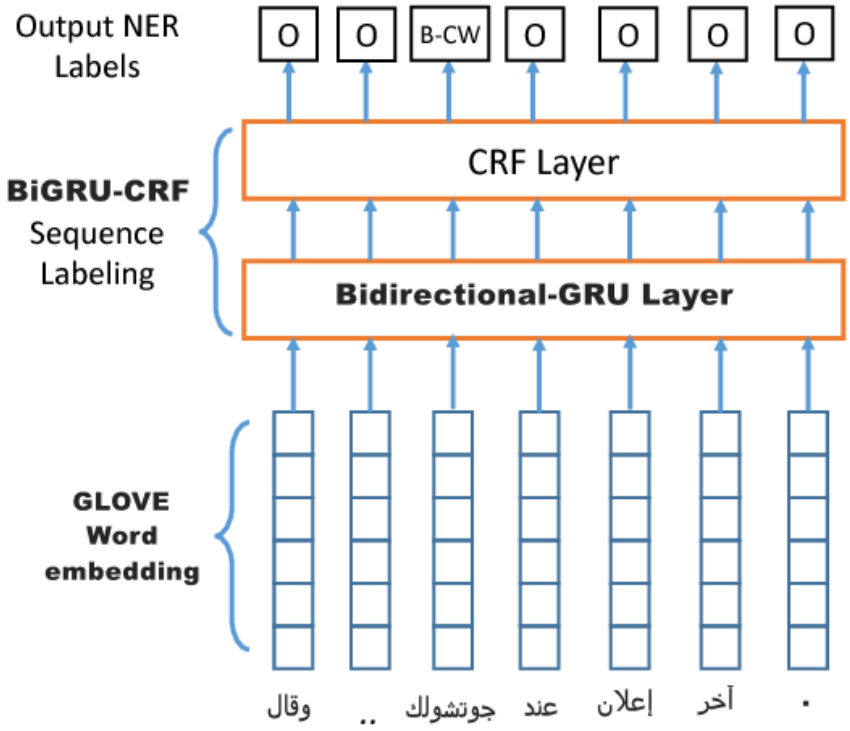

Fig. 2. GLOVE-GRU-CRF network architecture in the proposed method

In the third model, which is called FastText-CNN-CRF, the input layer is the embedding vectors obtained from the FasTetxt method, and the next layer is a one-dimensional CNN neural network, and at the end, a CRF layer is placed to predict NER labels. Present model framework is illustrated in Fig. 3 [50].

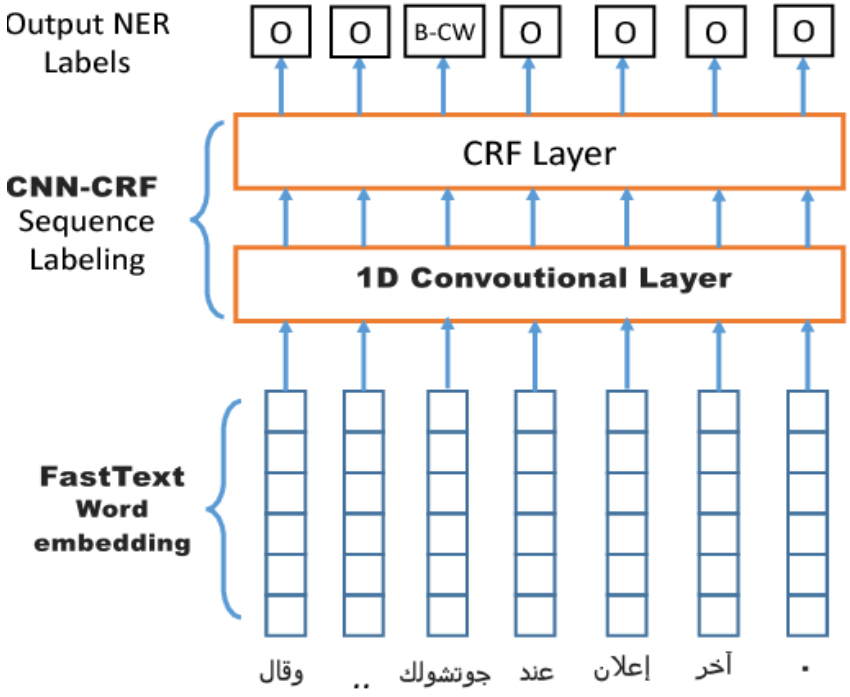

Fig. 3. FastText-CNN-CRF network architecture in the proposed method

3.4. Combination of Results from Three Models Using Majority Vote

In the last step of the proposed method, the final NER label for each token of the input text is obtained from the majority vote of the NER labels obtained from the GLOVE-GRU-CRF, W2V-LSTM-CR, and FastTtxt-CNN-CRF models, as shown in Fig. 4. it has been shown.

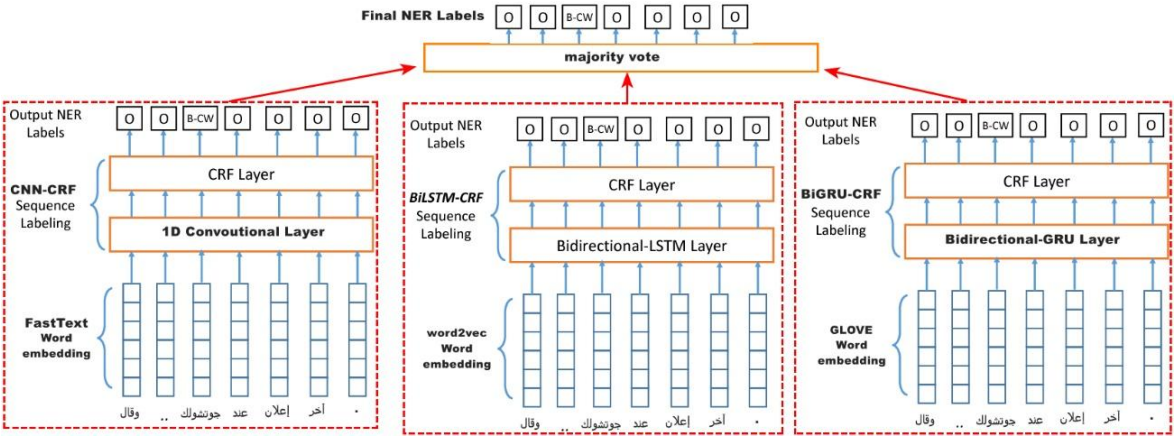

Fig. 4. Combination of results from three models using majority vote in the proposed method

4. Experiment and Result

We divided dataset in two collections of training and test units containing 80% and 20% of the dataset, respectively.

4.1. Dataset

We apply annotated set of data [51] for training and assessing scheme of ANERCorp. It is manually annotated ANE corpus which are generally accessible for the usage in objectives pursuit. It includes 32,114 named entities and 150,286 tokens from 316 articles which were chosen from several publications for making corpus as wide as feasible. Applying labelling IOB, corpus was annotated applying technique of annotation listed in MUC-6. For annotation, vocabs below with 9 classes used: B-ORG, B-MISC, I-ORG, I-LOC, B-PERS, O, I-PERS, I-MISC, B-LOC. PERS shows NE; LOC shows location; ORG shows company; MISC shows miscellaneous that refers to NE however unassociated with other classes; O shows extra vocabs which are not NEs. While the document is shared in 2 rows—1 for vocabs and the other for labels— training document kind as CONLL is applied. 20.6% of named entities were shared for organizations, 39% for individuals, 10% for others, and 30.4% for locations, as shown in Table 1.

Table 1. The ANERCorp dataset's training, validation, and testing statistics

	PER	LOC	ORG	MISC	O	Total
Train	5144	4121	2751	1319	96263	109598
Test	725	487	391	164	12361	14128
Validation	568	423	266	172	10559	11988
Total	6437	5031	3408	1655	119183	

4.2. Evaluation Criteria

Here, we apply Accuracy (Acc), recall (R), precision (P), and F1-score (F1), to assess our presented model performance. Allow TP to show tag numbers where a happening group is predicted to be accurate, FP to show tag numbers that wrongly predict other groups in such groups, and FN to show tag numbers that have not been recognized successfully. A completed formula is illustrated in (1)-(4):

$$Acc = \frac{TP + FP}{TP + TN + FP + FN} \quad (1)$$

$$R = \frac{TP}{TP + FN} \quad (2)$$

$$P = \frac{TP}{TP + FP} \quad (3)$$

$$F1 = \frac{2 * P * R}{P + R} \quad (4)$$

In Equation (1), Acc is shared accurate predictions amount by whole predictions amount. In Equation (2), *R* scales ability of system to detect entire NEs in provided corpus. In Equation (3), P quantifies options' precision and properness diagnosed by NER system. Since there is trade-off among recall as well as accuracy. In Equation (4), *F1* is used for balancing antagonistic associations between them.

4.3. Experiment Setting

In such test, Python was used for performing model, whole experiment was done on platform of Google Colab (https://colab.research.google.com/) with Tesla T4 GPU.

Tests are carried out, this is performed just on a dataset of ANERCorp. Four entities' assessment outcomes details recognized by the presented model are illustrated in Table 2. 'Acc', 'P', 'R', 'F', and 'J' respectively show accuracy, precision, recall, F1-score, and Jaccard, as shown in Table 2.

Table 2. The performance results of the proposed model

	Acc	R	P	F1score	J
LogisticRegression	0.903560	0.903560	0.903560	0.903560	0.824086
GaussianNB	0.222386	0.222386	0.222386	0.222386	0.125104
MLPClassifier	0.903472	0.903472	0.903472	0.903472	0.823939
DecisionTree	0.903472	0.903472	0.903472	0.903472	0.823939
SGDClassifier	0.903560	0.903560	0.903560	0.903560	0.824086
SVM	0.903472	0.903472	0.903472	0.903472	0.823939
RandomForest	0.903472	0.903472	0.903472	0.903472	0.823939
AdaBoost	0.889219	0.889219	0.889219	0.889219	0.800536
GradientBoosting	0.903060	0.903060	0.903060	0.903060	0.823253
XGBClassifier	0.898554	0.898554	0.898554	0.898554	0.815795
LGBMClassifier	0.903118	0.903118	0.903118	0.903118	0.823351
CatBoost	0.903560	0.903560	0.903560	0.903560	0.824086
LSTMW2V	0.938793	0.938793	0.938793	0.938793	0.884646
GRUGLOVE	0.950201	0.950201	0.950201	0.950201	0.905127
CNNFT	0.891851	0.891851	0.891851	0.891851	0.804812
Proposed method	**0.943883**	**0.943883**	**0.943883**	**0.943883**	**0.893730**

As can be seen in Table 2, the proposed model has significantly better performance compared to various models.

According to the results in Table 2, the suggested model performed significantly better than previous models; however, a thorough investigation of the mistakes is required to better understand the model's shortcomings and opportunities for improvement. The suggested model outperformed other models in terms of accuracy and F1-score. However, error analysis can assist discover specific trends that lead to prediction errors.

1. False Positives: Some entities may be wrongly classified as entities due to semantic similarities or textual attributes. These flaws are particularly noticeable in simpler models like

LogisticRegression and GaussianNB, which have lesser accuracy and analytical power. For example, the model may falsely identify some words or phrases as entities that do not appear in the names of persons, places, or organizations.

2. False Negatives: Complex models like LSTMW2V and GRUGLOVE have fewer false negatives, indicating superior performance. These models can detect entities more effectively in a variety of scenarios, although they may occasionally neglect certain entities. For example, the model may be unable to correctly identify the names of certain locations or organizations that are mentioned indirectly or deeply in the text. This could be owing to the inaccuracy of neural network-based models like LSTM or GRU, which, in some cases, require additional data for further learning.
3. The effect of employing the conditional random field (CRF) layer: The results suggest that the proposed model, which combines the CRF layer with the BiLSTM, BiGRU, and CNN models, improves entity recognition accuracy significantly. This indicates that the model fared better at analyzing sentence structure and semantic context. As a result, the majority of false negative mistakes are caused by complicated semantic contexts in which the model need more information to recognize items.

4.4. Discussion

To evaluate the results, we compare provided model performance compared to state-of-the-art baselines. For provided paper comparison with [9], [19]-[21]. Furthermore, we used ANERCorp splits set of data provided by writers when available and standard 80% as dataset of training, 20% as dataset of testing. The presented model performance is contrary to the first [9], the second [19] the third [20], and the fourth baseline performance is shown in Table 3. In other words, Table 3 show the presented model performance contrary to the other baselines, in turn, on a dataset of ANERCorp.

Table 3. Comparison of ANER models performance

Model approach	Accuracy	Recall	Precision	F1score
[9]	-	94.28	93.22	93.74
[19]	-	82	84.7	83.3
[20]	-	90.54	93.52	92.01
[21]	91.24	78.33	88.33	83
Proposed method	94.39	94.39	94.39	94.39

Table 3 shows that the presented model performed better [9] by 0.65 F1-score points, indicating that the pre-trained model's context semantic representation of actively created vectors of words is superior to non-contextual representations of word vectors to show sentence features. Based on the outcomes illustrated in Table 3, adding a layer of CRF for BiLSTM, BiGRU, and CNN model's joint decoding obtained important developments over other models for NER on whole metrics. Provided scheme outperformed [19] by 11.09 points that reflects carrying info benefit illustrated by provided scheme to DL model to aid learning more context knowledge. Furthermore, since this is shown in Table 3, our model outperformed [20] by 2.38 F1-score points. In other words, the presented model performed better [21] by 11.39 F1-score points, by combining the CRF layer with a model of BiLSTM, BiGRU, and CNN also using glove, fast text, and word2 vec as models of text representation, the presented model performance was later increased.

This study investigates the effect of various word indexing algorithms, including GloVe, Word2Vec, and FastText, as well as deep learning architectures such as BiLSTM, BiGRU, and CNN, on the performance of the entity recognition model.

- Word2Vec: Preserving semantic links improves model accuracy in detecting things in basic phrases.
- GloVe: Using global correlations enhanced model accuracy in complicated sentences with deeper meaning.

- FastText: Using subwords increased the model's ability to identify unfamiliar and difficult words.

On the other hand, using advanced designs like:

- BiLSTM model improved understanding of complicated semantic relationships by processing input from both directions.
- BiGRU, like BiLSTM, fared well in complicated text processing due to its simpler structure and faster speed.
- The CNN model's capacity to recognize local features and semantic patterns improved its performance in basic texts.

The combination of these methodologies and architectures, particularly with the CRF layer, dramatically improved the model's accuracy in detecting things and its capacity to analyze complicated and unfamiliar input.

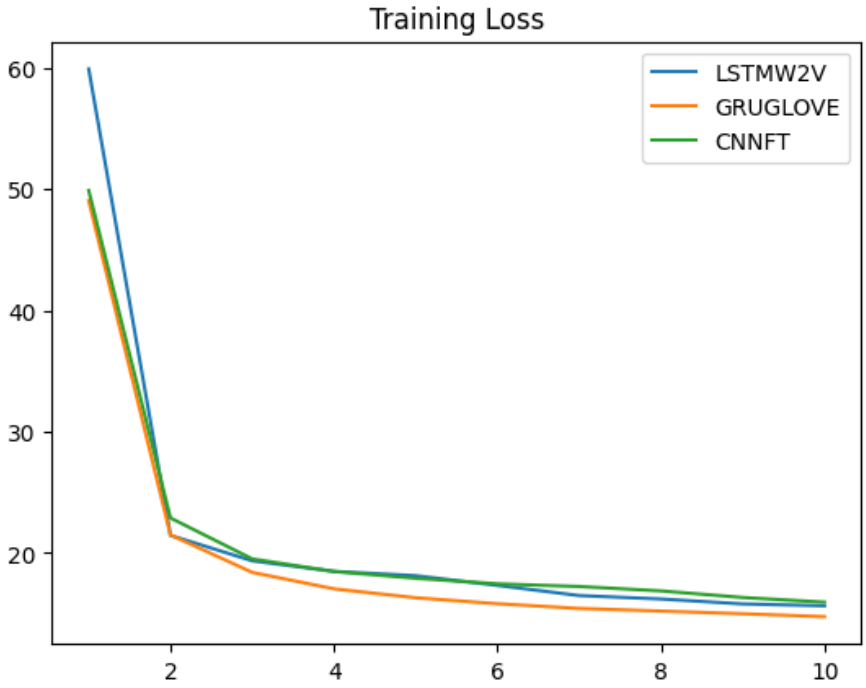

Fig. 5. Best-performing models' validation and training loss (a) LSTMW2V, (b) GRUGLOVE, (c) CNNFT

The performance of training in terms of training loss, is achieved by various networks at 10 epochs. Fig. 5 shows the training loss across 10 iterations for the whole best-performing network models on the ANERCorp set of data.

The models' training times vary according to the architecture's complexity and the type of word indexing approach used. BiLSTM and BiGRU-based models require more training time than simpler models like Logistic Regression and Decision Tree, but offer superior performance. More complicated word indexing methods, such as GloVe and FastText, required more memory and took longer to process, whereas Word2Vec consumed fewer resources. Combining deep learning architectures with a CRF layer increased memory and processing time, but the higher computational cost was mitigated by a considerable gain in model accuracy.Finally, the suggested model requires moderate to high computing resources, which for resource-constrained contexts might be optimized by lowering the number of model layers or adopting lighter indexing approaches.Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

5. Conclusion

In this paper, a hybrid model for Arabic Named Entity Recognition was presented, which outperformed previous simple models. However, this approach has shortcomings that must be addressed in future studies. One of the key constraints is the reliance on the ANERCorp dataset, which is limited in terms of linguistic diversity and domain coverage. This can impact the model's capacity to generalize to new data and domains. Furthermore, despite its high performance, the suggested model is computationally demanding and requires substantial processing resources, particularly during training. This may be a limitation when implementing the model in low-resource environments or real-time systems. Furthermore, error analysis revealed that the model makes false positive and false negative errors when identifying specific entities or cases with semantic ambiguity, necessitating future modifications to the model structure. Future study should test the model on a broader and more diverse dataset to determine its generalizability. Model optimization techniques and computational complexity reduction can also help enhance performance in resource-constrained contexts. Using more complex methodologies, such as Transformer-based models, and investigating their impact on system accuracy and efficiency, can potentially provide new opportunities for increasing model performance.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] X. Qu, Y. Gu, Q. Xia, Z. Li, Z. Wang and B. Huai, "A Survey on Arabic Named Entity Recognition: Past, Recent Advances, and Future Trends," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 3, pp. 943-959, 2024, https://doi.org/10.1109/TKDE.2023.3303136.

[2] I. Keraghel, S. Morbieu, and M. Nadif, "Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study," *Computation and Language*, 2024, https://doi.org/10.48550/arXiv.2401.10825.

[3] T. E. Moussaoui and C. Loqman, "Advancements in Arabic Named Entity Recognition: A Comprehensive Review," *IEEE Access*, vol. 12, pp. 180238-180266, 2024, https://doi.org/10.1109/ACCESS.2024.3491897.

[4] R. Salah, M. Mukred, L.Q. binti Zakaria, and F.A.M. Al-Yarimi, "A Machine Learning Approach for Named Entity Recognition in Classical Arabic Natural Language Processing," *KSII Transactions on Internet and Information Systems*, vol. 18, no. 10, pp. 2895-2919, 2024, http://dx.doi.org/10.3837/tiis.2024.10.005.

[5] Z. Hu, W. Hou, and X. Liu, "Deep learning for named entity recognition: a survey," *Neural Computing and Applications*, vol. 36, pp. 8995–9022, 2024, https://doi.org/10.1007/s00521-024-09646-6.

[6] M. Abedi, L. Hempel, S. Sadeghi, and T. Kirsten, "GAN-Based Approaches for Generating Structured Data in the Medical Domain," *Applied Science*, vol. 12, no. 14, p. 7075, 2022, https://doi.org/10.3390/app12147075.

[7] R. Anam *et al*., "A deep learning approach for Named Entity Recognition in Urdu language," *PLoS ONE*, vol. 19, no. 3, p. e0300725, 2024, https://doi.org/10.1371/journal.pone.0300725.

[8] M. N. A. Ali, G. Tan and A. Hussain, "Boosting Arabic Named-Entity Recognition With Multi-Attention Layer," *IEEE Access*, vol. 7, pp. 46575-46582, 2019, https://doi.org/10.1109/ACCESS.2019.2909641.

[9] E. Çano and M. Morisio, "Word Embeddings for Sentiment Analysis: A Comprehensive Empirical Survey," *Computation and Language*, 2019, https://doi.org/10.48550/arXiv.1902.00753.

[10] F. Almeida and G. Xexéo, "Word embeddings: A survey," *Computation and Language*, 2023, https://doi.org/10.48550/arXiv.1901.09069.

[11] A. Allahim and A. Cher, "Advancing Arabic Word Embeddings: A Multi-Corpora Approach with Optimized Hyperparameters and Custom Evaluation," *Applied Science*, vol. 14, no. 23, p. 11104, 2024, https://doi.org/10.3390/app142311104.

[12] K. Ullah, A. Rashad, M. Khan, Y. Ghadi, H. Aljuaid, Z. Nawaz, "A Deep Neural Network-Based Approach for Sentiment Analysis of Movie Reviews," *Complexity*, vol. 2022, no. 1, pp. 1-9, https://doi.org/10.1155/2022/5217491.

[13] S. F. Sabbeh and H. A. Fasihuddin, "A Comparative Analysis of Word Embedding and Deep Learning for Arabic Sentiment Classification," *Electronics*, vol. 12, no. 6, p. 1425, 2023, https://doi.org/10.3390/electronics12061425.

[14] V. -I. Ilie, C. -O. Truică, E. -S. Apostol and A. Paschke, "Context-Aware Misinformation Detection: A Benchmark of Deep Learning Architectures Using Word Embeddings," *IEEE Access*, vol. 9, pp. 162122-162146, 2021, https://doi.org/10.1109/ACCESS.2021.3132502.

[15] N. Alsaaran and M. Alrabiah, "Arabic Named Entity Recognition: A BERT-BGRU Approach," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 471–485, 2021, https://doi.org/10.32604/cmc.2021.016054.

[16] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for Arabic language understanding," *Computation and Language*, 2020, https://doi.org/10.48550/arXiv.2003.00104.

[17] C. Helwe, G. Dib, M. Shamas, and S. Elbassuoni, "A Semi-Supervised BERT Approach for Arabic Named Entity Recognition," *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pp. 49-57, 2020, https://aclanthology.org/2020.wanlp-1.5/.

[18] C. Helwe, G. Dib, M. Shamas, and S. Elbassuoni, "A Semi-Supervised BERT Approach for Arabic Named Entity Recognition," *Seminar Slides*, 2020, https://a3nm.net/work/seminar/slides/20210204-helwe.pdf.

[19] H. Mahdhaoui, A. Mars, and M. Zrigui, "Active Learning with AraGPT2 for Arabic Named Entity Recognition," *Advances in Computational Collective Intelligence*, pp. 123–135, 2023, https://doi.org/10.1007/978-3-031-41774-0_18.

[20] H. Mahdhaoui, A. Mars, and M. Zrigui, "Building the ArabNER Corpus for Arabic Named Entity Recognition Using ChatGPT and Bard," *Intelligent Information and Database Systems*, pp. 159–170, 2024, https://doi.org/10.1007/978-981-97-4982-9_13.

[21] H. Nayel, N. Marzouk and A. Elsawy, "Named Entity Recognition for Arabic Medical Texts Using Deep Learning Models," *2023 Intelligent Methods, Systems, and Applications (IMSA)*, pp. 281-285, 2023, https://doi.org/10.1109/IMSA58542.2023.10217658.

[22] Z. Zheng, Y. Cang, W. Yang, Q. Tian, and D. Sun, "Named Entity Recognition: A Comparative Study of Advanced Pre-trained Models," *Journal of Computer Technology and Software*, vol. 3, no. 5, 2024, https://doi.org/10.5281/zenodo.136240.

[23] N. Alsaaran and M. Alrabiah, "Classical Arabic Named Entity Recognition Using Variant Deep Neural Network Architectures and BERT," *IEEE Access*, vol. 9, pp. 91537-91547, 2021, https://doi.org/10.1109/ACCESS.2021.3092261.

[24] H. Wei *et al*., "Named Entity Recognition From Biomedical Texts Using a Fusion Attention-Based BiLSTM-CRF," *IEEE Access*, vol. 7, pp. 73627-73636, 2019, https://doi.org/10.1109/ACCESS.2019.2920734.

[25] A. Shaker, A. Aldarf, and I. Bessmertny, "Using LSTM and GRU with a New Dataset for Named Entity Recognition in the Arabic Language," *Computation and Language*, 2023, https://doi.org/10.48550/arXiv.2304.03399.

[26] S. Kumar-Birthriya, P. Ahlawat, and A. K. Jain, "Enhanced Phishing Website Detection Using Dual-Layer CNN and GRU with Attention Mechanism and Lexical NLP Features," *SN Computer Science*, vol. 5, p. 929, 2024, https://doi.org/10.1007/s42979-024-03282-6.

[27] H. Mahdhaoui, A. Mars and M. Zrigui, "Optimizing Arabic Named Entity Recognition through Active Learning and AraBERT," *2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1-5, 2023, https://doi.org/10.1109/INISTA59065.2023.10310315.

[28] A. Chaimae, E. Y. Yacine, M. Rybinski and J. F. A. Montes, "BERT for Arabic Named Entity Recognition," *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pp. 1-6, 2020, https://doi.org/10.1109/ISAECT50560.2020.9523676.

[29] S. Albahli, "An Advanced Natural Language Processing Framework for Arabic Named Entity Recognition: A Novel Approach to Handling Morphological Richness and Nested Entities," *Applied Sciences*, vol. 15, no. 6, p. 3073, 2025, https://doi.org/10.3390/app15063073.

[30] N. Alshammari, S. Alanazi, "An Arabic dataset for disease named entity recognition with multi-annotation schemes," *Data*, vol. 5, no. 3, p. 60, 2020, https://doi.org/10.3390/data5030060.

[31] M. Al-Smadi, S. Al-Zboon, Y. Jararweh and P. Juola, "Transfer Learning for Arabic Named Entity Recognition With Deep Neural Networks," *IEEE Access*, vol. 8, pp. 37736-37745, 2020, https://doi.org/10.1109/ACCESS.2020.2973319.

[32] H.-S. Le, T.-V. H. Do, M. H. Nguyen, H.-A. Tran, T.-T. T. Pham, N. T. Nguyen, and V.-H. Nguyen, "Predictive Model for Customer Satisfaction Analytics in E-commerce Sector Using Machine Learning and Deep Learning," *International Journal of Information Management Data Insights*, vol. 4, no. 2, p. 100295, 2024, https://doi.org/10.1016/j.jjimei.2024.100295.

[33] A. Youssef, M. Elattar and S. R. El-Beltagy, "A Multi-Embeddings Approach Coupled with Deep Learning for Arabic Named Entity Recognition," *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 456-460, 2020, https://doi.org/10.1109/NILES50944.2020.9257975.

[34] M. Zhang, G. Geng, and J. Chen, "Semi-Supervised Bidirectional Long Short-Term Memory and Conditional Random Fields Model for Named-Entity Recognition Using Embeddings from Language Models Representations," *Entropy*, vol. 22, no. 2, p. 252, 2020, https://doi.org/10.3390/e22020252.

[35] A. B. Sadallah, O. Ahmed, S. Mohamed, O. Hatem, D. Hesham and A. H. Yousef, "ANER: Arabic and Arabizi Named Entity Recognition using Transformer-Based Approach," *2023 Intelligent Methods, Systems, and Applications (IMSA)*, pp. 263-268, 2023, https://doi.org/10.1109/IMSA58542.2023.10217635.

[36] K. Abainia, "DZDC12: A New Multipurpose Parallel Algerian Arabizi–French Code-Switched Corpus," *Language Resources and Evaluation*, vol. 54, pp. 419–455, 2020, https://doi.org/10.1007/s10579-019-09454-8.

[37] M.N.A. Ali and G. Tan, "Bidirectional Encoder–Decoder Model for Arabic Named Entity Recognition," *Arabian Journal for Science and Engineering*, vol. 44, pp. 9693–9701, 2019, https://doi.org/10.1007/s13369-019-04068-2.

[38] B. A. Benali, S. Mihi, N. Laachfoubi, A. A. Mlouk, "Arabic named entity recognition in arabic tweets using bert-based models," *Procedia Computer Science*, vol. 203, pp. 733-738, 2022, https://doi.org/10.1016/j.procs.2022.07.109.

[39] A. Mousa, I. Shahin, A. B. Nassif and A. Elnagar, "Cascaded RBF-CBiLSTM for Arabic Named Entity Recognition," *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pp. 1-5, 2020, https://doi.org/10.1109/CCCI49893.2020.9256638.

[40] F. Genuario, G. Santoro, M. Giliberti, S. Bello, E. Zazzera, and D. Impedovo, "Machine Learning-Based Methodologies for Cyber-Attacks and Network Traffic Monitoring: A Review and Insights," *Information*, vol. 15, no. 11, p. 741, 2024, https://doi.org/10.3390/info15110741.

[41] A. Aldumaykhi, S. Otai and A. Alsudais, "Comparing Open Arabic Named Entity Recognition Tools," *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 46-51, 2023, https://doi.org/10.1109/IRI58017.2023.00016.

[42] O. Obeid *et al*, "CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing," *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 7022–7032, 2020, https://aclanthology.org/2020.lrec-1.868.

[43] G. Bourahouat, M. Abourezq, and N. Daoudi, "Word Embedding as a Semantic Feature Extraction Technique in Arabic Natural Language Processing: An Overview," *The International Arab Journal of Information Technology*, vol. 21, no. 2, pp. 313-325, 2024, https://doi.org/10.34028/iajit/21/2/13.

[44] A. Kutuzov, "Distributional Word Embeddings in Modeling Diachronic Semantic Change," *University of Oslo Library*, 2020, http://urn.nb.no/URN:NBN:no-84130.

[45] S. Helmstetter and H. Paulheim, "Collecting a Large Scale Dataset for Classifying Fake News Tweets Using Weak Supervision," *Future Internet*, vol. 13, no. 5, p. 114, 2021, https://doi.org/10.3390/fi13050114.

[46] C. Zhang *et al*., "From Word Vectors to Multimodal Embeddings: Techniques, Applications, and Future Directions for Large Language Models," *Computation and Language*, 2024, https://doi.org/10.48550/arXiv.2411.05036.

[47] I. Gagliardi and M.T. Artese, "Semantic Unsupervised Automatic Keyphrases Extraction by Integrating Word Embedding with Clustering Methods," *Multimodal Technologies and Interaction*, vol. 4, no. 2, p. 30, 2020, https://doi.org/10.3390/mti4020030.

[48] B. Yu and Z. Fan, "A Comprehensive Review of Conditional Random Fields: Variants, Hybrids and Applications," *Artificial Intelligence Review*, vol. 53, pp. 4289–4333, 2020, https://doi.org/10.1007/s10462-019-09793-6.

[49] T. Mayer, E. Cabrio, and S. Villata, "Transformer-Based Argument Mining for Healthcare Applications," *ECAI 2020*, vol. 325, 2020, https://doi.org/10.3233/FAIA200334.

[50] E. Dayanik, "Challenges of Computational Social Science Analysis with NLP Methods," *Online Publications of University Stuttgart*, 2022, https://doi.org/10.18419/opus-12530.

[51] M. Al-Duwais, H. Al-Khalifa, and A. Al-Salman, "A Benchmark Evaluation of Multilingual Large Language Models for Arabic Cross-Lingual Named-Entity Recognition," *Electronics*, vol. 13, no. 17, p. 3574, 2024, https://doi.org/10.3390/electronics13173574.