

# A Survey on Arabic Named Entity Recognition: Past, Recent Advances, and Future Trends

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, Baoxing Huai

**Abstract**—As more and more Arabic texts emerged on the Internet, extracting important information from these Arabic texts is especially useful. As a fundamental technology, **Named entity recognition (NER)** serves as the core component in information extraction technology, while also playing a critical role in many other Natural Language Processing (NLP) systems, such as question answering and knowledge graph building. In this paper, we provide a comprehensive review of the development of Arabic NER, especially the recent advances in deep learning and pre-trained language model. Specifically, we first introduce the background of Arabic NER, including the characteristics of Arabic and existing resources for Arabic NER. Then, we systematically review the development of Arabic NER methods. Traditional Arabic NER systems focus on feature engineering and designing domain-specific rules. In recent years, deep learning methods achieve significant progress by representing texts via continuous vector representations. With the growth of pre-trained language model, Arabic NER yields better performance. Finally, we conclude the method gap between Arabic NER and NER methods from other languages, which helps outline future directions for Arabic NER.

**Index Terms**—Named entity recognition, Arabic texts, Deep learning, **Pretrained language model**

## 1 INTRODUCTION

ARABIC is the official language in the Arab world and one of the richest natural languages in the world in terms of morphological inflection and derivation. It is among top 10 most popular languages in the world with 420M native speakers, and more than 25 popular dialect [1]. Arabic has many unique characteristics that make understanding of Arabic text both challenging and interesting. Thanks to the increasing availability of social media platforms, the amount of Arabic texts available on the Internet has seen a significant leap. To effectively process these texts, Arabic Named Entity Recognition (ANER) attracts increasing attention. Named entity recognition (NER) is the task of identifying mentions that correspond to specific types, such as person name, location, and organization. Besides general domains, in several specific domains, such as the medical field, drugs, and clinical procedures can also be extracted by NER. With Arabic NER, it is convenient to extract important information in different domains. NER can be used for different downstream tasks, such as relation extraction [2], entity linking [3], event extraction [4], coreference resolution [5], and machine translation [6].

As a basic component of natural language processing (NLP), NER is well-studied in English but still lacks comprehensive exploration in Arabic. What makes English NER easier than Arabic NER is that most of the mentions in English begin with capital letters which is not an option in Arabic. In addition, Arabic is a morphologically complex language due to its extremely inflectional and the existence of many variations. There are three categories: Classical Arabic (CA), Modern Standard Arabic (MSA) and Dialects

Cite	Year	Type	Title
[8]	2009	Report	NERA: Named entity recognition for Arabic
[9]	2014	<b>Survey</b>	A survey of Arabic named entity recognition and classification
[10]	2015	Report	Named entity recognition for arabic social media
[11]	2016	<b>Survey</b>	Arabic named entity recognition —a survey and analysis
[12]	2017	<b>Survey</b>	A comparative review of machine learning for Arabic named entity recognition
[13]	2019	Report	Arabic named entity recognition using deep learning approach
[14]	2019	Report	Arabic named entity recognition: What works and what's next
[15]	2020	<b>Survey</b>	A recent survey of arabic named entity recognition on social media.

TABLE 1: Existing surveys and reports for Arabic NER. “Report” denotes a competition report or describes a detailed system implementation. “Survey” means a comprehensive investigation of different methods.

Arabic (DA) [7]. CA is the original Arabic language that has been used for over 1,500 years, and it is usually used in most Arabic religious texts. MSA is one of the official languages of the United Nations and widely used in today’s Arabic newspapers, letters, and formal meetings, which is also focused by researchers. DA is spoken Arabic used in informal daily communication.

The development of solutions for Arabic NER has evolved according to advancements in algorithms and computational power. As to the techniques applied in Arabic NER, early NER systems depended on handcrafted rules. Afterward, machine learning approaches were trained on hand-crafted features to classify words into different named entity classes. Subsequently, deep learning techniques that

- X. Qu, Y. Gu, Q. Xia, Z. Li, Z. Wang, B. Huai are with Huawei Cloud Language and Speech Innovation Lab, Hangzhou 310051, China. E-mail: {quxiaoye, guyingjie4, xiaqingrong, lizechang1, wangzhefeng, huaibaoxing}@huawei.com  
Zhefeng Wang is the corresponding author.

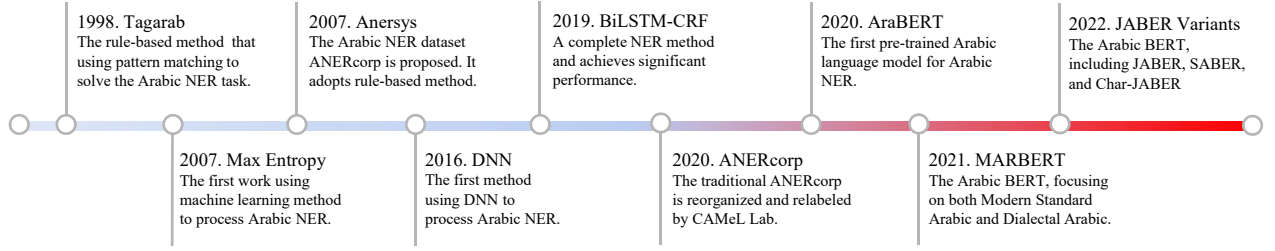


Fig. 1: Milestones in the development of Arabic NER.

rely on word embeddings and automatically discover representations for entity classification became dominant. More recently, the transformer-based pre-trained language model further promotes the development of this field with huge computational power. In this paper, we mainly focus on the development of named entities in Arabic. We do not claim this article to be exhaustive of all NER works on all languages.

**Motivation for conducting this survey.** Over the past few years, a large number of methods have explored deep learning methods in Arabic NER and achieved decent performance. We notice these methods adopt similar networks, mainly optional Bi-LSTM for encoding and CRF for decoding. However, there are much more abundant solutions in English and we hope these methods can also benefit the development of Arabic NER. Meanwhile, to the best of our knowledge, there are few systematic surveys in this field. As shown in Table 1, we list previous reports and surveys in Arabic NER. Shaalan et al. [9] conducted a survey of Arabic NER in 2014 but focused on handcrafted features. Dandashi et al. [11] and Salah et al. [12] mainly explore the machine learning method for Arabic NER. More recently, Ali et al. [15] studied the Arabic NER on social media in 2020 but performed experiments on English datasets. In summary, existing surveys do not completely cover modern DL-based NER systems. In Table 1, there are several reports introducing the competition method or system architecture, such as [14] which is the champion of Arabic Named Entity Recognition challenge run by Topcoder.com. These reports describe an effective system but do not provide a comprehensive analysis of the development of Arabic NER. There are also recent NER surveys focusing on deep learning technology [16], [17] in general domains in English. These surveys do not consider the language characteristic and specific challenges in Arabic. Thus, a comprehensive review of existing NER methods in Arabic is missing. We hope this review can help sort out the past Arabic NER methods systematically (as shown in Figure 1). It is worth noticing that discovering valuable information in a less rich language, such as Arabic is also a significant chance for the NLP community.

**Contribution for conducting this survey.** We summarize the contributions as below:

- To the best of our knowledge, this is the first comprehensive survey focusing on recent advances in Arabic NER, including the Arabic NER corpus, current resources, rule-based methods, machine learning methods, deep learning methods, and up-to-date pre-trained language model methods. In addition, we also sur-

vey the most representative methods for applied deep learning techniques in Arabic NER problems.

- In this survey, due to the unfair comparison of previous methods from different data split, we carefully summarize the results of Arabic NER on the ANERCorp and AQMAR datasets. In this way, the researchers can easily compare different methods. We believe the fair comparison will contribute to more competitive Arabic NER methods.
- We outline the potential directions for Arabic NER, including the alternative solutions for Arabic NER, fine-grained and nested Arabic NER, low-resource and distantly-supervised NER, and cross-linguistic knowledge transfer for Arabic NER. All these future trends will lead to more effective downstream applications.

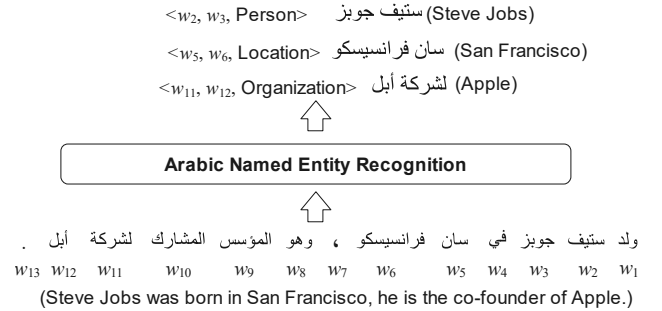


Fig. 2: An illustration of the Arabic NER task.

## 2 BACKGROUND

In this section, we first give a formal definition of Arabic NER. Subsequently, we introduce the challenges of Arabic NER. Finally, we collect the current resources for Arabic NER and describe the evaluation metrics.

### 2.1 What is Arabic NER

Arabic NER aims to locate and classify named entities in Arabic text  $s = \langle w_n, \dots, w_2, w_1 \rangle$  (Different from the left-to-right style of writing in Chinese or English, Arabic is writing from right to left) into predefined entity categories. Same as other common NER tasks, the Arabic NER can also be divided into general domain including Person, Location and Organization types, and specific domain (e.g. Biomedical, News and Finance).

There are two typical outputs for NER, one is a list of tuples  $\langle P_s, P_e, t \rangle$ , each of which is a mention of named

TABLE 2: List of annotated datasets for Arabic NER. “#Tags” refers to the number of entity types.

Corpus	Year	Text Source	#Tags	URL
ANERcorp	2007	Website, news, magazines	4	<a href="https://camel.abudhabi.nyu.edu/anercorp/">https://camel.abudhabi.nyu.edu/anercorp/</a>
ACE 2003	2004	Broadcast News, newswire genres	7	<a href="https://catalog ldc.upenn.edu/LDC2004T09">https://catalog ldc.upenn.edu/LDC2004T09</a>
ACE 2004	2004	Broadcast News, newswire genres	7	<a href="https://catalog ldc.upenn.edu/LDC2005T09">https://catalog ldc.upenn.edu/LDC2005T09</a>
ACE 2005	2005	Transcripts, news	7	<a href="https://catalog ldc.upenn.edu/LDC2006T06">https://catalog ldc.upenn.edu/LDC2006T06</a>
ACE 2007	2007	Transcripts, news	7	<a href="https://catalog ldc.upenn.edu/LDC2014T18">https://catalog ldc.upenn.edu/LDC2014T18</a>
REFLEX	2009	Reuters news	4	<a href="https://catalog ldc.upenn.edu/LDC2009T11">https://catalog ldc.upenn.edu/LDC2009T11</a>
AQMAR	2012	Arabic Wikipedia	7	<a href="http://www.cs.cmu.edu/~ark/ArabicNER/">http://www.cs.cmu.edu/~ark/ArabicNER/</a>
OntoNotes 5.0	2013	News	18	<a href="https://catalog ldc.upenn.edu/LDC2013T19">https://catalog ldc.upenn.edu/LDC2013T19</a>
WDC	2014	Wikipedia	4	<a href="https://github.com/Maha-J-Althobaiti/Arabic_NER_Wiki-Corpus">https://github.com/Maha-J-Althobaiti/Arabic_NER_Wiki-Corpus</a>
DAWT	2017	Wikipedia	4	<a href="https://github.com/klout/opendata/tree/master/wiki_annotation">https://github.com/klout/opendata/tree/master/wiki_annotation</a>
CANERCorpus	2018	Religion	14	<a href="https://github.com/RamziSalah/Classical-Arabic-Named-Entity-Recognition-Corpus">https://github.com/RamziSalah/Classical-Arabic-Named-Entity-Recognition-Corpus</a>
Wojood	2022	Wikipedia	21	<a href="https://ontology.birzeit.edu/Wojood/">https://ontology.birzeit.edu/Wojood/</a>

entity. Here  $P_s$  and  $P_e$  are the start and end positions of a named entity mention,  $t$  is the type of the mention from a predefined category set. Another one considers NER as a sequence labeling task, where each token is assigned to a special label (e.g. B-PER, I-PER, O). Figure 2 shows an example that an Arabic NER model recognizes three named entities from the given Arabic text.

## 2.2 The Challenges of Arabic NER

NER is especially challenging when it comes to Arabic because of its particularities and unique nature. The main characteristics of Arabic that pose non-trivial challenges for NER tasks are as follows:

- **No Capitalization:** Capitalization is a very distinguishing orthographic feature, such as proper names and abbreviations, which helps NER systems accurately recognize specific types of entities. Unlike English, where a named entity usually begins with a capital letter, capitalization is not a feature of the Arabic words. Consequently, only relying on looking up the items in the lexicon of proper nouns is not an appropriate solution to this problem [18].
- **The Agglutinative Nature:** Arabic has a high agglutinative nature in which a word may consist of prefixes, lemma, and suffixes in different combinations, and that leads to a very complicated morphology [19]. Meanwhile, named entities in Arabic are often associated with some other words, such as prepositions (e.g. for/to, as and by/with) or conjunctions (e.g. and) [9]. A common treatment is to obtain the text segmentation separated by a space character [20].
- **Spelling Variants:** In Arabic, the word may be spelled differently and still refers to the same word with the same meaning, creating a many-to-one ambiguity, which is also called transcriptional ambiguity [21]. To tackle this problem, researchers may normalize each variant to a unified form by a mechanism such as string distance calculation [22].
- **No Short Vowels:** Short vowels, or diacritics, are needed for pronunciation and disambiguation. However, most modern Arabic texts do not include diacritics. Therefore, one word form in Arabic may refer to two or more different words or meanings according

to the context they appear, leading to a one-to-many ambiguity, which is easy to understand for a native Arabic speaker, but not for an Arabic NER system [23]. Short vowels are very important because different diacritics represent different meanings, which could help to disambiguate the structure and lexical types of Arabic entities. For a long time, these ambiguities caused by short vowels can only be solved by using contextual information and some knowledge of the language [24]. Recently, with the introduction of the Arabic pre-trained models (e.g. AraBERT [25] and Ara-ELECTRA [26]), the challenge caused by diacritics is further alleviated.

- **Lacking Linguistic Resources:** Adequate data resources, including annotated corpus and high-quality lexicons/gazetteers, are critical to building a strong NER system. However, unlike the languages such as English and Chinese that have large-scaled tagged corpus, there is a limitation in the number of available Arabic linguistic resources for Arabic NER due to the absence of annotations. Even so, a few Arabic corpora have been created and publicly available for research purposes, we will introduce more details of these Arabic resources in the section 2.3.

## 2.3 Current Resources for Arabic NER

In this section, we summarize some widely-used datasets and popular tools for Arabic NER. As shown in Table 2, most datasets are mainly developed by annotating news articles with a small number of entity types. As most works of Arabic NER report their results on ANER, here we also give a short description of this dataset.

ANERcorp [27] consists of 316 articles from 11 diverse sources to acquire a generalized corpus and is composed of a training set and a test set. It contains 150,286 tokens. In addition to the traditional entities such as organization, location, and person, ANERcorp introduces the miscellaneous type to suggest a single class for entities that do not belong to any of the other classes. **It is worth noting that** this dataset was split into many different configurations over the years which made it hard to compare fairly across papers and systems. In 2020, a group of researchers from CAMeL Lab met with the creator of

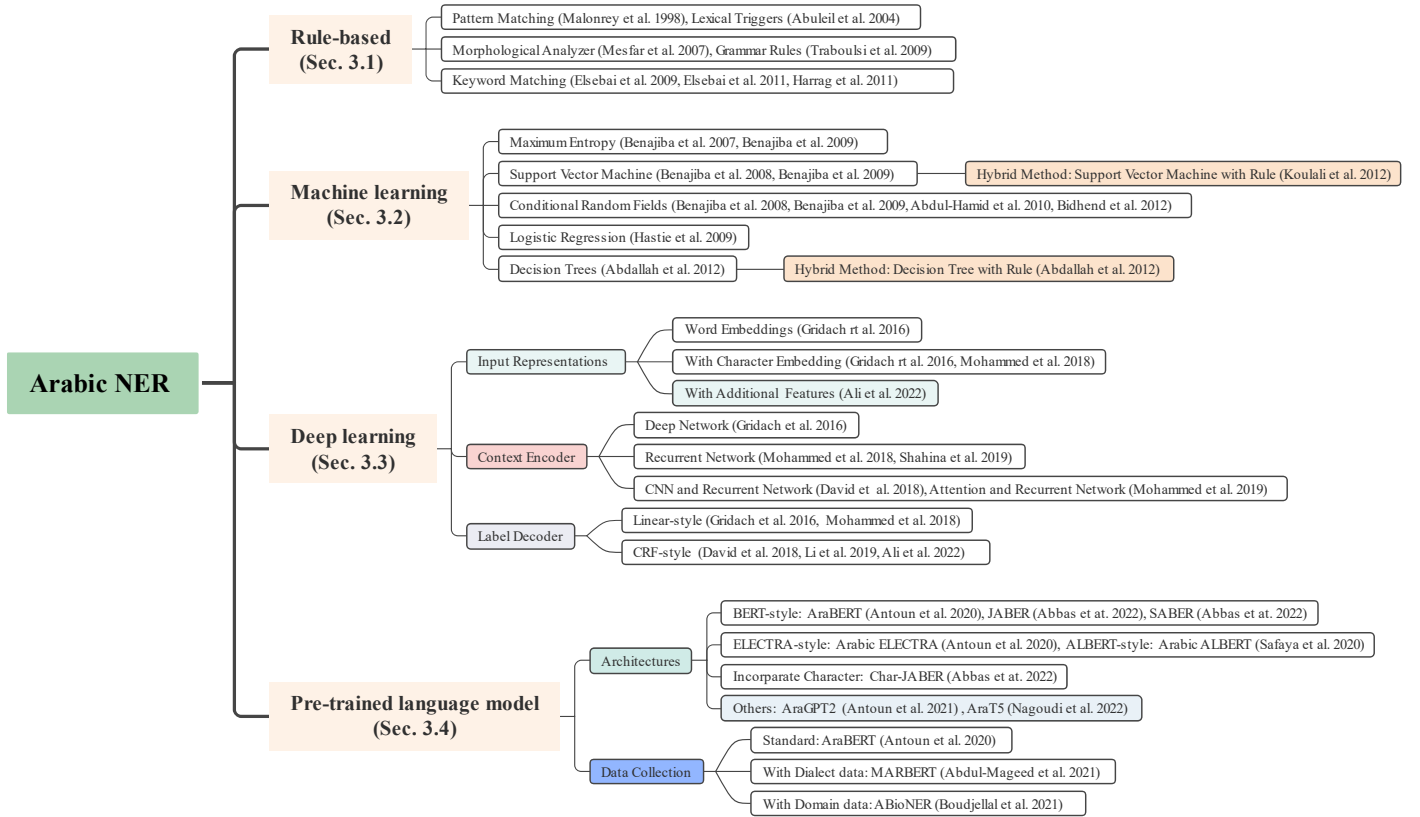


Fig. 3: The taxonomy of Arabic NER methods based on different methods. The rule-based method and machine learning method will be introduced in Sections 3.1 and 3.2. The methods of deep learning and pre-trained language model for Arabic NER will be described in Sections 3.3 and 3.4. In deep learning methods, we present the development of Arabic NER from three perspectives, including input representations, context encoder, and label decoder. In the pre-trained language model, we illustrate the Arabic NER methods according to their architectures and data collection scope.

the corpus to consult with him and collectively agree on an exact split, and accepted minor corrections from the original dataset. Thus, a new version is ANERcorp<sup>1</sup> is released and we suggest using this split as a standard dataset for evaluating NER performance in Arabic. In this new version of the dataset, the sentences containing the first 5/6 of the words go to train and the rest go to test. The train split has 125,102 words and the test split has 25,008 words.

Besides the dataset, here we also introduce some commonly used Arabic NLP tools which contribute to Arabic NER processing. Specifically, MADAMIRA [28]<sup>2</sup> provides morphological analysis and disambiguation of Arabic. It is designed to deal with basic tasks which are often the first or intermediate steps of a complex system, including diacritization, lemmatization, morphological disambiguation, part-of-speech tagging, stemming, glossing, and tokenization. The input text is converted into the Buckwalter representations and passed to the morphological analysis. Then the SVM extracts closed-class features while the language models acquire open-class features. The ranking component scores the analysis list of each word, which is subsequently passed to the tokenization component. Both the base phase chunking component and named entity recognizer take SVM to process the analyses and tokenizations.

The Stanford Arabic segmenter [29]<sup>3</sup> is an extension of the character-level conditional random field model. This segmenter expands the regular label space to introduce two Arabic-specific orthographic rules. Domain adaptation methods are applied to segment informal and dialectal Arabic text.

Farasa [30]<sup>4</sup> is a fast and accurate Arabic segmenter based on SVM which ranks possible segmentations of a word by adopting a variety of features and lexicons. Farasa supports segmentation, stemming, named entity recognition, part of speech tagging and diacritization. The experiments demonstrate that this tool performs much better on the information retrieval tasks than Stanford Arabic segmenter and MADAMIRA.

As most Arabic NLP tools suffer from a lack of flexibility, CaMel [31]<sup>5</sup> is presented to solve the problem. It is implemented in Python 3 and provides a series of pre-processing steps, such as transliteration and orthographic normalization. As for the named entity recognition, CaMel fine-tunes the AraBert and experiments show that this tool outperforms the CRF-based system.

1. <https://camel.abudhabi.nyu.edu/anercorp/>

2. <https://camel.abudhabi.nyu.edu/madamira/>

3. <https://nlp.stanford.edu/projects/arabic.shtml>

4. <https://farasa.qcri.org/>

5. [https://github.com/CAMEL-Lab/camel\\_tools](https://github.com/CAMEL-Lab/camel_tools)



## 2.4 Arabic NER metrics

The metrics we take for Arabic NER are either the exact-match ones or the relaxed-match ones. For exact-match metrics, we evaluate the approaches with precision, recall, and F-score. Both the boundaries and entity type must be correct if the predicted entity is regarded as accurate. The precision suggests the ability to extract correct entities while the recall measures the ability to find as many entities as possible. Accordingly, F-score represents the balance of precision and recall. However, the exact-match metrics are somewhat too strict. Some approaches are also evaluated using relaxed-match metrics. For example, MUC-6 [32] propose to evaluate the system through the entity type and ignore the boundaries as long as there is an overlap with ground truth boundaries. Since the relaxed-match metric is complicated and inconvenient to analyze the bad cases, they are still not widely used.

## 3 METHOD

To help reveal the development of Arabic NER, we present several representative methods. As shown in Figure 3, the Arabic NER methods are summarized as four paradigms, including the rule-based methods, the machine learning methods, deep learning methods, and recent pre-trained language model methods. In the following sections, we will describe the important methods in each paradigm.

### 3.1 Rule-based

Before machine learning and deep learning methods are widely applied in NER tasks, the rule-based approach is most commonly used. Rule-based NER systems [8], [33]–[37] depend on manually compiled local grammatical rules extracted and verified by experts in linguistics. Gazetteers and lexicons (or dictionaries) are utilized as grammar rules in the context where the named entities appear. Basically, most of the words in gazetteers are used as trigger words to help identify a named entity within the given text.

There are only a few works that leverage rule-based methods in named entity recognition tasks in Arabic because encoding human professional knowledge requires a lot of work from expert linguists, and usually just works for a single language. At a very early stage of Arabic NER, Maloney and Niv [38] proposed a rule-based pattern-matching method for Arabic NER tasks named TAGARAB. This system supports the recognition of Name, Date, Time and Numeric entities by combining high-precision morphological analysis and a pattern matching engine.

However, the TAGARAB only considers the morphology of the Arabic, which is a low-level rule feature. Hence Traboulsi et al. [39] discussed the use of methods based on the so-called local grammar formalism in untagged Arabic corpora, in conjunction with a finite state automata to recognize patterns of person names in Arabic news texts. The results show that considering all the common grammars can save a lot of time and effort. Besides, Elsebai et al. [40] adopted a rule-based method making use of the Buckwalter Arabic Morphological Analyser (BAMA) [41] as well as using a set of keywords to analyze the texts that probably include named entities of person name, which achieved an F-measure of 89%.

More recently, Harrag et al. [42] proposed a finite state transducer-based entity extractor to identify key entities from prophetic narrations texts. Concretely, they first adopted finite state transducer to assign a conceptual label among a finite set of labels to capture different types of named entities. Then for each detected concept, the goal of the system is to fill a pattern that contains the event description. The model achieved a 52.00% F-measure on a set of prophetic narrations texts.

To sum up, rule-based NER methods are easy to understand based on solid linguistic knowledge. However, the shortcomings of rule-based systems are also noticeable. First of all, any maintenance or updates required for these rule-based systems is labor-intensive and time-consuming. Secondly, the problem is compounded if the linguists with the required knowledge and background are not available.

### 3.2 Machine Learning

Generally speaking, machine learning-based NER methods [43] utilize ML algorithms to learn entity tagging decisions. They treat the NER problem as a classification task and require the availability of a large annotated dataset.

Aiming at tackling Arabic NER tasks based on machine learning methods, Benajiba et al. ([20], [27], [44], [45]) did a lot of valuable works in the early stage. They first adopted Maximum Entropy (ME) [27] and developed an Arabic NER system, ANERSys 1.0. The features used by the system are lexical, contextual and gazetteers features. However, this system is hard to recognize entities with more than one word. Subsequently, the developed ANERSys 2.0 [20] was proposed.

In order to further improve the performance of the Arabic NER system, Benajiba et al. [44] considers more features such as morphological, POS-tags and so on. They tested the new NER system on ACE Corpora and ANERcorp and obtained the state-of-the-art performance when all the features are used. Conditional Random Fields (CRF) based method is a strong baseline machine learning method [46]. Benajiba et al. [45] further used CRF for improving ANERSys based on ME approach. The results showed that using the CRF method, the model can achieve nearly two percentage points better performance than ANERSys 2.0. Other machine learning technologies, such as Logistic Regression [47] and Hidden Markov Models (HMM) are also widely used in the Arabic hybrid NER system and could further improve the performance compared with rule-based systems.

Besides solely relying on machine learning algorithms, a few methods [37], [43], [48] propose to integrate the rule-based approach and machine learning-based approach together to create a hybrid system with the aim of enhancing the overall performance of Arabic NER tasks. Specifically, the rule-based approach relies on handcrafted grammatical rules, while ML-based approach takes advantage of the ML algorithms that utilize sets of features extracted from datasets for building NER systems. Then, the hybrid approach combines rule-based approach with ML-based approach together in a pipeline process to improve the overall performance of the system. For instance, Abdallah et al. [37] combines the Decision Tree with rule to recognize

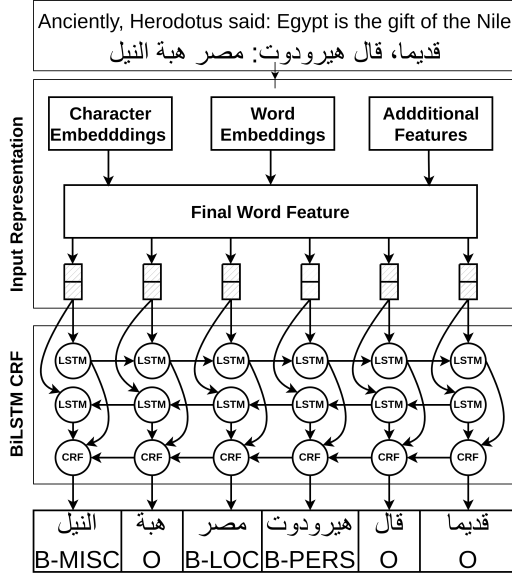


Fig. 4: Architecture of a representative deep learning method [49] for Arabic NER. This model is a typical BiLSTM-CRF architecture and adopts multiple input features.

three types of named entities including person, location, and organization, and proved its comparatively higher efficiency as a classifier in the hybrid NER system. It is worth noting that the hybrid approach suffers from an increase in complexity because it involves the use of grammatical rules and handcrafted features.

Although the machine-learning methods achieve decent performance, they are not able to learn complex and high-level features from data through linear models such as log-linear HMM or linear chain CRF. To make up the deficiency of machine learning, deep learning methods are presented to discover hidden features automatically. We will introduce deep learning methods in section 3.3.

### 3.3 Deep Learning

Recently, deep learning methods have been widely proposed for Arabic NER. Compared to traditional handcraft feature based approaches, deep learning methods are able to learn more robust features and reduce the effort of designing handcraft features. Moreover, the deep learning methods can be trained in an end-to-end manner by gradient descent. Benefiting from this, we can design possibly more complex Arabic NER systems. In this section, as shown in Figure 3, we will introduce the development of deep learning methods in Arabic NER from three perspectives, namely the input representations, the context encoder, and the label decoder. The *input representations* consider word- and character-level embeddings as well as additional features which are effective for Arabic NER. *Context encoder* captures the context dependencies from the text using CNN, RNN, or other networks. *Label decoder* aims to predict corresponding labels for tokens in the input sequence.

To apply deep learning in Arabic NER, Gridach et al. [50] first introduce a simple deep neural network (DNN) for Arabic NER in 2016. Based on the word2vec [51] which represents words in low dimensional real-valued dense

vectors, they further construct three neural network layers, including an input layer, a hidden layer, and an output layer. In the input layer, they explicitly represent context as a window consisting of a center word concatenated with its immediate surrounding neighbors. The hidden layer learns feature combinations from the word representations of the window. Finally, the output layer transforms the hidden representations and classifies them into different entity types. This work adopts a simple linear layer to predict the label sequence, while the following works mostly utilize conditional random fields to determine the label sequence.

Subsequently, more complex context encoders are applied for Arabic NER, such as the Recurrent neural network (RNN). RNN effectively captures the contextual information for each token and has achieved remarkable success in many different NLP tasks [52], [53]. In Arabic NER, Mohammed et al. [54] adopted a bidirectional long short-term memory (BiLSTM) layer for entity recognition. The BiLSTM layer can efficiently use both past and future input features [55], which makes it effective for the NER task. Specifically, they first apply a character-level embedding layer and a word embedding layer as model inputs. Subsequently, they further propose an embedding attention layer to integrate these two kinds of embeddings. The integrated embeddings are then fed into the BiLSTM layers and the final classification layer. To evaluate the effectiveness of different recurrent networks, Shahina et al. [56] explore RNN, LSTM, and GRU to model the Arabic NER data where context of each word in the sequence is relevant.

As to input representations for Arabic NER, Gridach et al. [57] investigate the effectiveness of injecting character-level features based on the commonly used BiLSTM-CRF architecture. Specifically, they employ a BiLSTM encoder to encode the characters and concatenate the first and last character representations as the final character-level representation for the corresponding word. This character representation is then concatenated with the word embedding as the input to the NER model encoder. Besides, this paper shows that using pre-trained word embeddings outperforms randomly initialized word embeddings by a large margin. Finally, Gridach et al. [57] evaluate the proposed model on the Twitter NER dataset [58] and the results show the effectiveness of introducing character-level information and pre-trained word embeddings.

To encode both word-level and character-level input representations, David et al. [59] also adopt the BiLSTM to encode the combination of character- and word-level representations. The word embedding is passed into a CNN layer before being fed into the BiLSTM. Mohammed et al. [60] propose an efficient multi-attention layer system for the Arabic NER task. Similar to the previous work, they adopt character-level embeddings and word embeddings. Differently, they further introduce a self-attention layer after encoding the embeddings with the BiLSTM, which can boost the performance of the system.

Besides traditional character embeddings and word embeddings, as shown in Figure 4, Ali et al. [49] explore additional word features from an external knowledge source. The basic architecture is also BiLSTM-CRF. They add morphological and syntactic features to different word embeddings to train the model. Specifically, they use the 2019

release of the Madamira tool [28] to generate the Part-of-Speech (POS), capitalization, and word analysis features. Furthermore, they design a fourth feature which is the quote feature. All features are represented as one-hot vectors and concatenated to the word and character embeddings.

Besides supervised training, Helwel et al. [61] propose a semi-supervised co-training approach to the realm of deep learning, named deep co-learning. Specifically, they use a small amount of labeled data with a large amount of partially labeled data from Wikipedia. They first train two deep learning classifiers based on the fully-labeled dataset and different sets of word embeddings. These two models serve as different views for partially labeled data. Subsequently, they iteratively retrain classifiers by applying each classifier on the partially labeled corpus and using the output of each classifier to re-train the other classifier.

Liu et al. [62] propose an integrated Arabic NER system that combines deep representation learning, feature engineering, sequence labeling, and model ensemble. Following the common practice, Liu et al. [62] treat Arabic NER as a sequence labeling problem and employ the BiLSTM-CRF architecture as one baseline. To boost the model performance, Liu et al. [62] introduce a dictionary-based method to enhance the model outputs. Specifically, they first build a named entity map from the surface words to their entity types. The map will be used in the prediction layer that adds extra entities if the surface words match some entities in the map and there is no conflict between the surface words and the model predictions. Experiments show that the dictionary-based method can improve the recall value at a relatively small cost of the precision value, therefore improving the F1 score. Besides, handcrafted features are also employed in this work that categorize the input tokens into three types: numbers, English words, and Arabic words and are mapped into vectors. Experimental results show a significant improvement of 4 F1 score on the AQMAR dataset.

Compared with rule-based and machine learning based NER methods, deep-learning-based Arabic NER shows its superior performance over the past decade. In addition, deep-learning-based Arabic NER does not require hand-crafted features, which simplifies the usage of Arabic NER.

### 3.4 Pre-trained Language Model

In recent years, pre-trained language models (PLMs) have shown superior representative abilities to capture contextual information in sentences, thus achieving significant improvements in a variety of NLP tasks, including NER. Generally, most PLMs employ the Transformer architecture as the encoder, which consists of two layers, i.e., a multi-head attention layer and a 2-layer feed-forward neural network. Figure 5 (right) shows the architecture of one transformer block, where Figure 5 (middle) shows the details of the multi-head attention [63] and Figure 5 (left) depicts the inner scaled dot-product attention. The most important key elements to achieve a strong PLM depend on efficient model architectures and high-quality data collection. Thus, we will describe the Arabic PLM and its applications to Arabic NER from these two perspectives.

BERT [64] is one of the most famous pre-trained language models that uses masked language model (MLM)

and next sentence prediction (NSP) as the learning objective. For MLM, a subset of the input tokens are masked and the objective is to recover the original token at the masked position. The NSP task allows BERT to learn the relationship between two sentences by predicting whether the second sentence is the true next or not. By pre-training on large-scale unlabeled datasets, BERT learns rich text information and can provide powerful contextual information that facilitates downstream tasks. However, most of the previous PLM works focus on English, with little attention to the Arabic language. In this section, we mainly review PLM works for Arabic, demonstrating the effectiveness of Arabic PLM for Arabic NER.

Antoun et al. [25] pre-trained an Arabic BERT named AraBERT for the purpose of achieving the same success as BERT for English. Following BERT, Antoun et al. [25] also employ MLM and NSP as the training objective. To accommodate the characteristic of Arabic language, the authors propose to segment words into stems, prefixes, and suffixes first with an offline segmenter. The pre-trained model is evaluated on three Arabic language understanding tasks, including sentiment analysis, named entity recognition, and question answering.

Antoun et al. [26] build AraELECTRA, whose model architecture is the same as ELECTRA [65]. The entire model contains two neural network modules, i.e., 1) a generator that takes the corrupted sentences as input and learns to predict the original tokens that have been masked and 2) a discriminator that takes as input the predicted sentence and try to tell which tokens are masked and which tokens are original. In the fine-tuning stage, the input sentence is fed into the discriminator with an additional linear layer for downstream tasks. They evaluated the Arabic ELECTRA model on the ANERCorp dataset and the results reveal its superiority.

Besides training objectives, the lightweight pre-trained model also attracts increasing attention for a faster inference speed. Following English ALBERT [66], Safaya et al. [67] pre-trained an Arabic ALBERT with Arabic texts. The main difference between BERT and ALBERT is that ALBERT has the same parameter across different layers, thus largely reducing the model parameter size. Considering the language characteristic of Arabic, Ghaddar et al. [68] propose a CharJABER which enhances the BERT-style Arabic PLM with character-level representations at the input layer.

There are also some other style Arabic PLMs except the encoder-style AraBERT, such as decoder-style AraGPT2 [69] and encoder-decoder-style AraT5 [70]. These works are promising for the development of Arabic NLP. However, since these models are not evaluated on Arabic NER tasks, we will not introduce them in detail.

Based on the above proposed Arabic PLMs, Norah et al. [71] propose the first work that tackles the classical Arabic NER. They propose to fine-tune a BiLSTM/BiGRU-CRF model on Arabic BERT and compare with two simple baselines, i.e., 1) a linear layer with softmax function on top of BERT (BERT) and 2) a CRF layer on top of BERT (BERT-CRF). They conduct experiments on the classic Arabic NER dataset CANERCorpus. The experimental results show that the two simple baselines (BERT, BERT-CRF) have already achieved promising results of 94.45 and 94.68 F1

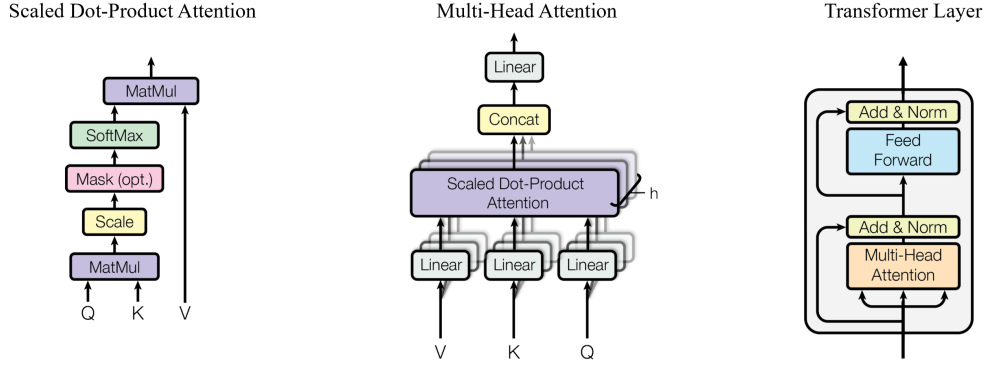


Fig. 5: The architectures of dot-product attention, multi-head attention, and transformer layer from left to right. The figure is from [63]. They are core components in pre-trained language models.

score, respectively. The BERT-BiGRU-CRF model achieves the best result of 94.76 F1 score.

To further explore the effectiveness of different PLMs architectures, Al-Qurishi et al. [72] propose a Transformer-based CRF model for Arabic NER that fine-tuned on three BERT variants, i.e., AraBERT, AraELECTRA, and XLM-Roberta. The experiments are conducted on the ANERCorp and AQMAR datasets and the results show that fine-tuning on AraBERT achieves the best performance.

Apart from the fully supervised methods for Arabic NER, Helwe et al. [73] propose a semi-supervised framework based on a teacher-student learning mechanism, called deep co-learning. The basic Arabic NER backbone is based on the AraBERT. In the training process, they first train a teacher model with the labelled dataset. Then, the trained teacher model is used to predict the pseudo labels on the partially labelled dataset. Finally, the result pseudo dataset is used to train a student model. However, the trained model has lower performance on the TWEETS dataset [74] compared with the other two datasets, i.e., AQMAR and NEWs dataset [74]. The reason may come from that the AraBERT is trained mainly on the MSA corpus while the TWEETS dataset mostly belongs to the Egyptian dialect. This phenomenon highlights another important component for Arabic NER, i.e., high-quality data collection.

To extend the Arabic PLMs to diverse data domains and different Arabic dialects, Abdul-Mageed et al. [75] collect the training data from several sources, i.e., books, new articles, common crawl data, and Wikipedia and train an Arabic PLM MARBERT. To promote the understanding of the Arabic dialect, they also train the model with a larger Twitter dataset. The Twitter dataset is collected from a large in-house dataset of 6 billion tweets, resulting in a collection of 128GB texts. In the training process, this work removes the next sentence prediction task since the tweets are usually short. The extensive experiments perform on five NER datasets, i.e., ACE03NW, ACE03BN, ACE04NW, ANERcorp, and TW-NER. We notice that MARBERT achieves a new SOTA on TW-NER, verifying the effectiveness of pre-training with tweets.

In addition to the commonly used news-domain NER dataset, several works also try to promote the progress of the Arabic NER in the biomedical domain. Biomedical Arabic NER is more challenging due to its significant

domain divergence. For this reason, the typical Arabic PLMs, such as AraBERT, usually perform not satisfactory. To solve this problem, Boudjellal et al. [76] propose an Arabic biomedical pre-trained model by injecting biomedical knowledge into the AraBERT pre-trained model. They manually collect a set of literature corpora from medical journals, such as PubMed, MedlinePlus Health Information. The collected corpus contains about 500k words, which is much smaller compared with the data used to train AraBERT. Even though, the experimental results show that the trained biomedical model achieves better performance than AraBERT on the bioNER data. This result also demonstrates the value of training a domain-specific pre-trained model for the corresponding domain-specific task.

Up to now, Arabic NER has achieved significant progress with the surge of pre-trained language models. On the one hand, existing Arabic PLMs provide powerful representations for Arabic texts that effectively boost the Arabic NER performance. On the other hand, more Arabic PLMs for specific domains are still urgently needed for Arabic NER from corresponding domains.

### 3.5 Discussion

After reviewing the four paradigms for Arabic NER, here we organize the experiment results on Arabic NER datasets in the order of publish date for further comparison. As shown in Table 3, we conclude the experimental results on ANERCorp. Due to the difference in data split strategies between methods, it is not easy to fairly compare them. In other words, the Arabic NER lacks a standard leaderboard, while the ANERCorp already has the training, development, and test set.

From these tables, we can observe that most traditional methods, including rule-based methods and machine learning methods, perform significantly worse than deep learning methods. For deep learning methods, they mainly adopt CRF for tag decoding and utilize LSTM for representation encoding. It is a common choice to encode the character representation by a CNN network and then integrate the character-level feature with the word-level feature. When it comes to pre-trained language model, the model performance achieves a significant leap. As shown in Table 3, comparing different pretrained model AraELECTRA, MARBERT, Arabic-BERT, ARBERT, and JABER, we can observe



TABLE 3: Comparison of Arabic NER systems on ANERCorp. The results are reported on four kinds of entities, including person, location, organization and miscellaneous. “Standard” in the Dataset Split means the methods use the official dataset split instead of splitting the dataset by themselves. As traditional rule-based and machine learning methods do not have a specific context encoder, we merge columns “Input representations” and “Context encoder”. “-” symbols in the table indicate the corresponding item is not described in the original paper.

Work	Year	Input Representations and Context Encoder	Label Decoder	Dataset Split	Performance (F1 score)
[27]	2007	N-Gram	Maximum Entropy	Standard	54.11
[45]	2008	-	CRF	Standard	65.91
[77]	2010	N-Gram	CRF	4:0:1	PER: 81, LOC: 88, ORG: 73
[48]	2012	-	SVM-HMMs	Standard	PER: 94.4, LOC: 96.9, ORG: 80.66, MISC: 60.82
[78]	2012	-	Rule	Standard	67.13
[79]	2013	-	CRF	3:0:1	68.05
[80]	2014	-	Rule, Machine Learning	Standard	PER: 94, LOC: 90, ORG: 88
[81]	2015	N-Gram	HMM	-	PER: 79, LOC: 78, ORG: 67
[50]	2016	Word2vec, DNN	Linear	Standard	88.64
[82]	2018	-	Rule	Standard	PER: 78, LOC: 67, ORG: 63
[54]	2018	Word2vec, CNN-based Character	-	8:1:1	87.12
[83]	2019	Word2vec, CNN-based Character, CNN, LSTM	CRF	8:1:1	88.77
[84]	2019	-	Rule	Standard	PER: 83, LOC: 92, ORG: 89
[13]	2019	Bi-LSTM	CRF	-	90.6
[25]	2020	AraBERT v1	Linear	-	81.9
[25]	2020	AraBERT v0.1	Linear	Standard	84.2
[85]	2020	AraBERT v0.1, BiGRU	Linear	8:1:1	90.51
[31]	2020	-	-	5:0:1	83
[86]	2021	-	-	5:0:1	81.39
[26]	2021	AraELECTRA	Linear	5:0:1	83.95
[68]	2022	MARBERT	Linear	5:0:1	80.5
[68]	2022	Arabic-BERT	Linear	5:0:1	82.05
[68]	2022	ARBERT	Linear	5:0:1	84.03
[68]	2022	JABER	Linear	5:0:1	84.20

that JABER obtains the best result 84.20. We believe this result can further improve as JABER uses a simple linear layer for label decoding.

### 3.6 Applied Techniques For Arabic NER

Previous sections outline typical network architectures for Arabic NER. In this section, we survey recently applied techniques that are being explored for Arabic NER.

#### 3.6.1 Deep Multi-Task Learning for Arabic NER

Multi-task learning [87], [88] has been widely applied in many tasks in natural language processing [89], including named entity recognition. In multi-task learning, we generally choose several similar tasks with different models, and the hidden layers of these models are partly shared. In this way, models can provide better representations that are suitable for all tasks and avoid overfitting.

For multi-task learning in Arabic, Jarrar et al. [90] convert the nested Arabic NER into a multi-task paradigm and **present a multi-task model to predict entities on Wojood**, a large-scale Arabic NER dataset collected from multiple sources including web articles, archives, and social media, covering both MSA and dialect. Specifically, they adopt AraBert [25] as the shared encoder and design different linear layers for each kind of entity type, which is used to predict corresponding BIO tags. In this way, the model can extract entities of different types independently. However, **they still can not solve the nested entities from the same type**. Eventually, this method can achieve an overall F1 of 88.40 on Wojood.

In addition, Ahmed et al. [91] combine the Arabic NER task with the topic modeling task. They accordingly present

a dataset Tafsir as the first large-scale multi-task benchmark for Classical Arabic. These two tasks share the same embeddings. Then, they adopt a BiLSTM-CRF model for the upper layer NER task while using the linear layer for each label of the topic modeling task. Experiments show that the proposed method works effectively on both tasks.

#### 3.6.2 Transfer Learning for NER

Low-resource languages often suffer from the scarcity of training data, which can be effectively alleviated by transfer learning [92], [93]. Especially, transfer learning on pre-trained models can benefit many NLP tasks due to the knowledge transferred across tasks or domains.

For Arabic NER, to leverage the knowledge transfer ability from the multilingual pre-trained model, Al-Smadi et al. [94] propose a Pooled-GRU network based on Multilingual Universal Sentence Encoder [95] (MUSE) as shown in Figure 6. Specifically, they adopt a traditional sequence labeling model and a pre-trained multilingual universal sentence encoder to perform Arabic NER. In this way, the knowledge from the MUSE is well transferred to the Pooled-GPU. Benefiting from the transferred knowledge, the semantic relationships among sentences and phrases in other linguistic scenarios can be applied in Arabic as well. Experiments are conducted on an Arabic dataset WikiFANE [96] and this model demonstrates significant improvement compared with the BiLSTM-CRF model.

Besides the above multilingual MUSE, there are other multi-lingual transformers designed for cross-lingual tasks. Lan et al. [97] present GigaBert, a bilingual Bert model, which is proposed for Arabic NLP tasks and English-to-Arabic transfer. This customized Bert is pre-trained on Giga-

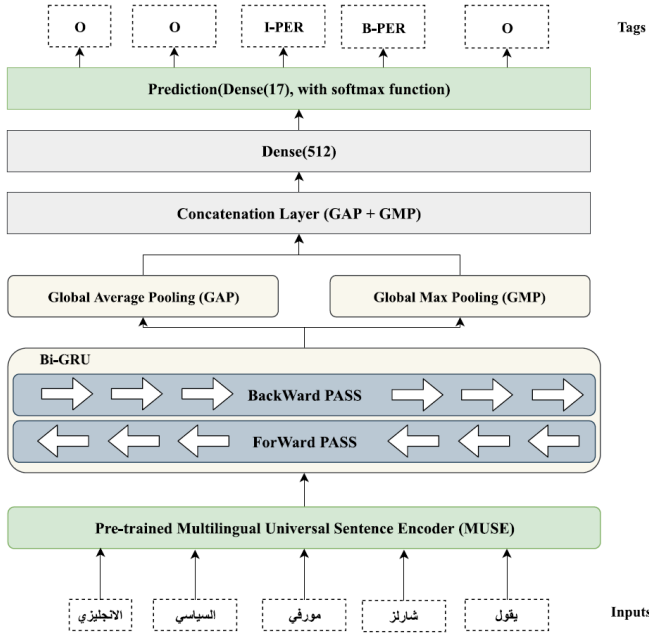


Fig. 6: Architecture of Pooled-GRU Model [94].

word and Wikipedia and some web crawl data. Specifically, the author customizes the vocabulary and creates the code-switched data to further strengthen the cross-lingual transfer ability. In this way, the model outperforms the popular multilingual Bert models in many NLP tasks including NER. In addition, with supervised Arabic data, GigaBert performs better than the state-of-art Arabic-specific Bert model.

#### 4 FUTURE DIRECTION

With the development of modeling methods and the increase in downstream applications, we expect Arabic NER to achieve more progress in the future. Thus, we list the potential further directions for Arabic NER from the following five perspectives. 1) The NER methods in English or Chinese have achieved significant progress in the past few years. The developments mainly focus on designing different label decoding strategies to adapt to diverse label structures. These important technology breakouts could be applied to Arabic NER but are not well explored. 2) Besides innovative technologies, particular Arabic NER tasks need to be further investigated by the requirement of the downstream applications, for instance, the nested NER. The nested NER is well studied in English but only a few works of Arabic NER consider this task. 3) Arabic NER, as a task with a small amount of available annotated data, also demands low-resource or distantly-supervised methods to relieve the problem of lacking data. In this way, Arabic NER can achieve a more robust performance under a few labeled annotated data. 4) Apart from leveraging more data to obtain a more robust Arabic NER model, cross-linguistic knowledge transfer also provides comprehensive information for model learning. To this end, we will analyze what cross-linguistic knowledge transfer technologies may benefit Arabic NER. 5) By jointly training NER with other auxiliary tasks, both two tasks can benefit from each other.

Thus, we also discuss several tasks interrelated with Arabic NER. We will describe these topics in the following sections.

**Optional Label Decoding Strategies.** As described in Section 3.3, nearly all methods for Arabic NER adopt linear classification or CRF layer for label decoding. As shown in Figure 7(a) and Figure 7(b), the linear layer or CRF decodes the label for each token. Although these two decoding strategies achieved decent performance in flat NER, they fail to process more complex NER tasks, such as nested NER. Different from these two paradigms, actually, in English or Chinese NER, there are many different label decoding strategies for NER as depicted in Figure 7(c)-(f). These newly proposed frameworks can unify different forms of NER tasks. Here we will simply introduce these representative solutions as follows:

- **MRC**<sup>6</sup> As shown in Figure 7(c), Li et al. [98] convert the NER task into a machine reading comprehension paradigm by inserting a question ahead of the sentence. The essence of using the MRC framework is to guide the model to search for the most relevant answers from the given text by designing NER label-related questions. During decoding, they identify the corresponding start and end positions for the type of entity mentioned by the question.
- **Biaffine**<sup>7</sup> In Figure 7(d), Yu et al. [99] recast the label decoding process into a dependency parsing task, where all pairs of start and end positions in the sentence are jointly considered and classified into specific types. The biaffine architecture is very effective and a lot of recent works have used it to further improve their performance [100], [101].
- **Locate-Label**<sup>8</sup> As depicted in Figure 7(e), different from previous one-stage paradigms, Shen et al. [102] propose a two-stage solution that first locates the potential entities in the sentence and then identifies the specific class of these spans. This framework can effectively utilize the boundary information of entities. Xia et al. [103] also adopt similar methods to process nested NER.
- **Generate**<sup>9</sup> In addition to the above encoder-based NER paradigms, recent encoder-decoder based methods attract increasing attention. As demonstrated in Figure 7(f), Lu et al. [104] transform the NER task into a generation task based on the generative pre-trained language model, such as T5 [105] or BART [106]. In this way, the model does not need to predict the entity class for each token but directly generates the entity in the decoded output sentence. Besides generating entity words, there are also generative methods [107], [108] which generate the indexes of entities in the original texts. Benefiting from the superior generative ability, these methods are unified for all kinds of NER tasks, including flat, nested, and discontinuous NER. It is worth noting that the performance of this paradigm may be heavily affected by user-constructed hard prompts. Therefore, how to design an appropriate prompt is also a challenge for Arabic NER.

6. <https://github.com/ShannonAI/mrc-for-flat-nested-ner>

7. <https://github.com/juntaoy/biaffine-ner>

8. <https://github.com/tricktreat/locate-and-label>

9. <https://github.com/universal-ie/UIE>

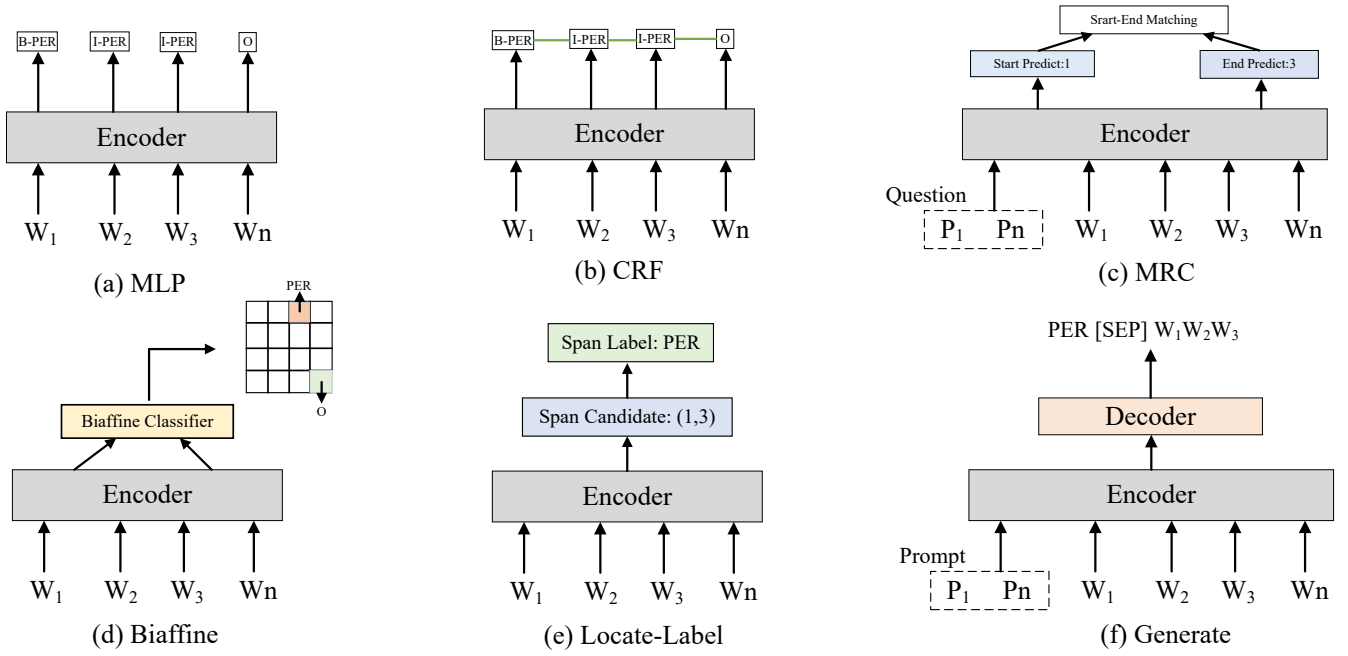


Fig. 7: Architectures of different paradigms of NER solutions. (a) and (b) are adopted in previous Arabic NER methods. (c)-(f) are recent representative decoding strategies in English NER.

**Particular NER Tasks.** As described in the method section, most existing methods for Arabic NER only consider coarse-grained NER types in the general domain and focus on flat NER. Actually, in real applications, such as slot filling [109]–[111] in conversation, the entities are fine-grained and hierarchical, which means a named entity may belong to multiple types. Another challenge is that the entity could be nested, where a part of one entity corresponds to another entity type. In addition, the entities in the text may also be discontinuous, especially in the clinical corpus. Under this setting, researchers need to move beyond continuous entities and propose new methods to extract discontinuous ones. Thus, we can resort to mainstream methods of English NER to solve these challenges in Arabic NER. Here we briefly describe several solutions for these challenges.

- **Fine-grained NER.** To explore fine-grained NER where hundreds of named entity categories must be recognized, Mai et al. [112] utilize the dictionary and category embeddings incorporating the neural network. Peng et al. [113] propose to adjust the existing NER system using the previously labeled data and entity lexicons of the newly introduced entity types. In this way, they can quickly adapt to fine-grained entities.
- **Nested NER.** As shown in Table 4, although Arabic NER methods adopting CRF as a decoder achieves better performance than linear in flat NER datasets, CRF architecture is hard to work in nested NER. To solve the nested NER, we can use the above proposed alternative solutions for NER, such as MRC, Biaffine, Locate-Label or Generate. There are also several graph-based methods that build an entity-entity graph [114] or model the nested ner as latent lexicalized constituency parsing [115]. With these methods, we can solve Arabic nested NER in a more comfortable way.

- **Discontinuous NER.** Traditional NER is presented as a sequence-labeling task and hard to handle these entities. Li et al. [116] propose a span-based model and perform relation classification to determine whether a given pair of entity fragments are overlapping or succession. Wang et al. [117] build a segment graph for each sentence, in which each node denotes a segment and an edge connects two nodes that belong to the same entity. Thus, they can identify the discontinuous entities. There are also several transition-based method [118] or grid-tagging method [119] proposed for this task. Moreover, the generation-based model can directly output discontinuous entities.

**Low-resource and Distantly-supervised NER.** In fact, the open-source labeled dataset and available Arabic entity gazetteer of Arabic NER are very scarce. Labeling a large amount of NER data is both time-consuming and expensive. This predicament leads to unsatisfactory performance in practical applications of Arabic NER. To make the best use of limited labeled data, it is feasible to consider low-resource and distantly-supervised methods to achieve better results. Here we present several effective methods from English that may work for Arabic NER.

- **Low-resource NER.** There are many works that utilize data augmentation to improve low-resource performance. Dai et al. [120] use label-wise token replacement, synonym replacement, and mention replacement to obtain new training samples. Ding et al. [121] model the linearized labeled sentences to generate new examples. Liu et al. [122] present an augmentation approach that can elicit knowledge from BERT and does not need considerable human intervention. With the augmentation method, we can obtain much more training examples. Besides generating more training samples, there

are also several recent methods [123], [124] focusing on designing low-resource algorithms. They effectively leverage the knowledge from the large pre-trained language models and achieve significant results.

- **Distantly-supervised NER.** With the help of Arabic gazetteer or existing NER tools, we can easily obtain abundant weak-labeled samples by matching or labeling unlabeled data. However, such weak labels are noisy. To comprehensively utilize these noisy samples, Yang et al. [125] propose a partial annotation learning paradigm. More recently, [126] introduces PU-learning to label positive tokens instead of directly learning the weak labels. Teacher-student frameworks [127]–[130] are widely adopted to refine noisy labels and achieve superior performance. To jointly learn from a small amount of labeled data and large weakly labeled data, Jiang et al. [131] propose a multi-stage computational framework to achieve robust learning.

**Cross-linguistic Knowledge Transfer.** Transferring knowledge from high-resource language to low-resource language is proven effective to alleviate the data scarcity problem. To enhance the capabilities of the Arabic NER system, it is worth considering knowledge transfer from other languages. To perform cross-linguistic knowledge transfer, here we present several effective solutions.

- **Machine Translation.** Currently, over 100 languages can be supported by strong machine translation systems. Machine translation is a straightforward way to translate training sentences and entities from other languages into target languages. Thus, machine translation can be leveraged to improve performance to cross-lingual NER [120], [132], [133].
- **Adversarial Training.** A language-adversarial task during finetuning can significantly improve the cross-lingual transfer performance [134]. In other words, the adversarial techniques encourage the model to align the representations of English documents and their translations of target languages. Furthermore, the adversarial learning framework can leverage the unlabeled data and improve the result [135], where a discriminator picks less language-dependent target language data according to the similarity to the source language.
- **Multilingual Pretrained Language Model.** The multilingual pre-trained language models, such as mBERT, XLM [136], mT5 [137], provide a straightforward way to enable zero-shot learning via cross-lingual transfer, thus eliminating the need for labeled data for the target task and language [138]. For pre-trained languages, the cross-lingual transfer is especially efficient [139].

**Jointly training NER with other tasks.** There are many tasks that are semantically related to NER. It is worth exploring approaches for jointly performing NER and other tasks. Here we introduce most related tasks.

- **NER and Entity Linking.** Most entity linking systems discard mention detection, performing only entity disambiguation of previously detected mentions. Thus, the dependency between the entity linking and NER is ignored. In real scenarios, we can do joint learning of NER and EL in order to leverage the information of both tasks at every decision [140]–[142]. Furthermore,

by having a flow of information between the computation of the representations used for NER and entity linking, we are able to improve the model.

- **NER and Relation Extraction.** Both NER and relation extraction aim to extract structured information from unstructured texts. The typical pipeline approach [143], [144] is to first identify entity mentions, and next perform classification between every two mentions to extract relations. However, this pipeline leads to the error propagation issue. To this end, an alternative and more recent approach is to perform these two tasks jointly [145], [146]. Thus, we can leverage the interaction between tasks, resulting in improved performance.

## 5 CONCLUSION

As the development of Arabic named entity recognition is significantly below the English counterpart, this survey aims to provide a comprehensive review for Arabic named entity recognition to help researchers understand this field. Specifically, we introduce the background of Arabic NER, especially the unique challenges of Arabic NER compared to English NER. Then, we summarize the current resources for Arabic NER, including the datasets and popular processing tools. With these resources, researchers can start their exploration easily and have a quick glance at Arabic NLP. To help straighten out the development of Arabic NER methods, we carefully categorize the Arabic NER methods into four paradigms, i.e., rule-based methods, machine learning methods, deep learning methods, and pre-trained language model methods. In these four paradigms, we present the most representative methods and give a description of their architectures. Furthermore, we summarize the applications of Arabic NER which promote the improvements of downstream tasks. Finally, we highlight several potential directions for Arabic NER. We hope this survey can provide handy guidance when designing Arabic NER models and attract more researchers to contribute to this field.

## REFERENCES

- [1] I. Guellil, H. Saàdane, F. Azouaou, B. Gueni, and D. Nouvel, “Arabic natural language processing: An overview,” *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 5, pp. 497–507, 2021.
- [2] Q. Cheng, J. Liu, X. Qu, J. Zhao, J. Liang, Z. Wang, B. Huai, N. J. Yuan, and Y. Xiao, “Hacred: A large-scale relation extraction dataset toward hard cases in practical applications,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 2819–2831.
- [3] Y. Gu, X. Qu, Z. Wang, B. Huai, N. J. Yuan, and X. Gui, “Read, retrospect, select: An mrc framework to short text entity linking,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, 2021, pp. 12 920–12 928.
- [4] T. Zhu, X. Qu, W. Chen, Z. Wang, B. Huai, N. J. Yuan, and M. Zhang, “Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph,” *arXiv preprint arXiv:2112.06013*, 2021.
- [5] K. Clark and C. D. Manning, “Improving coreference resolution by learning entity-level distributed representations,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 643–653.
- [6] A. Ugawa, A. Tamura, T. Ninomiya, H. Takamura, and M. Okumura, “Neural machine translation incorporating named entity,” in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3240–3250.

- [7] A. Elgibali, *Investigating Arabic: Current parameters in analysis and learning*. Brill, 2005, vol. 42.
- [8] 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.
- [9] K. Shaalan, "A survey of arabic named entity recognition and classification," *Computational Linguistics*, vol. 40, no. 2, pp. 469–510, 2014.
- [10] A. Zirikly and M. Diab, "Named entity recognition for arabic social media," in *Proceedings of the 1st workshop on vector space modeling for natural language processing*, 2015, pp. 176–185.
- [11] A. Dandashi, J. A. Jaam, and S. Fofou, "Arabic named entity recognition—a survey and analysis," in *Intelligent Interactive Multimedia Systems and Services 2016*. Springer, 2016, pp. 83–96.
- [12] R. E. Salah and L. Q. binti Zakaria, "A comparative review of machine learning for arabic named entity recognition," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 2, pp. 511–518, 2017.
- [13] I. El Bazi and N. Laachfoubi, "Arabic named entity recognition using deep learning approach," *International Journal of Electrical & Computer Engineering* (2088-8708), vol. 9, no. 3, 2019.
- [14] L. Liu, J. Shang, and J. Han, "Arabic named entity recognition: What works and what's next," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 2019, pp. 60–67.
- [15] B. A. B. Ali, S. Mihi, I. El Bazi, and N. Laachfoubi, "A recent survey of arabic named entity recognition on social media," *Rev. d'Intelligence Artif.*, vol. 34, no. 2, pp. 125–135, 2020.
- [16] 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, 2020.
- [17] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2145–2158.
- [18] S. Algahtani, *Arabic named entity recognition: a corpus-based study*. The University of Manchester (United Kingdom), 2012.
- [19] S. AbdelRahman, M. Elarnaoty, M. Magdy, and A. Fahmy, "Integrated machine learning techniques for arabic named entity recognition," *IJCSI*, vol. 7, no. 4, pp. 27–36, 2010.
- [20] 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 *IJCAI*, 2007, pp. 1814–1823.
- [21] K. Shaalan and H. Raza, "Person name entity recognition for arabic," in *Proceedings of the 2007 workshop on computational approaches to semitic languages: common issues and resources*, 2007, pp. 17–24.
- [22] R. Steinberger, "A survey of methods to ease the development of highly multilingual text mining applications," *Language resources and evaluation*, vol. 46, no. 2, pp. 155–176, 2012.
- [23] Alkharashi and Ibrahim, "Person named entity generation and recognition for arabic language," in *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 2009, pp. 205–208.
- [24] 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, 2009.
- [25] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 9–15.
- [26] —, "Araelectra: Pre-training text discriminators for arabic language understanding," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 2021, pp. 191–195.
- [27] Y. Benajiba, P. Rosso, and J. M. Benedíruiz, "Anersys: An arabic named entity recognition system based on maximum entropy," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2007, pp. 143–153.
- [28] A. Pasha, M. Al-Badrashiny, M. Diab, A. El Kholi, 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, pp. 1094–1101.
- [29] W. Monroe, S. Green, and C. D. Manning, "Word segmentation of informal Arabic with domain adaptation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 206–211. [Online]. Available: <https://aclanthology.org/P14-2034>
- [30] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for Arabic," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 11–16. [Online]. Available: <https://aclanthology.org/N16-3003>
- [31] O. Obeid, N. Zalmout, S. Khalifa, D. Taji, M. Oudah, B. Alhafni, G. Inoue, F. Eryani, A. Erdmann, and N. Habash, "Camel tools: An open source python toolkit for arabic natural language processing," in *Proceedings of the 12th language resources and evaluation conference*, 2020, pp. 7022–7032.
- [32] R. Grishman and B. Sundheim, "Message Understanding Conference- 6: A brief history," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. [Online]. Available: <https://aclanthology.org/C96-1079>
- [33] S. Abuleil and M. Evens, "Extracting names from arabic text for question-answering systems," in *RIAO*, 2004, pp. 638–647.
- [34] S. Mesfar, "Named entity recognition for arabic using syntactic grammars," in *Natural Language Processing and Information Systems: 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, Paris, France, June 27-29, 2007. Proceedings 12*. Springer, 2007, pp. 305–316.
- [35] K. Shaalan and H. Raza, "Arabic named entity recognition from diverse text types," in *International conference on natural language processing*. Springer, 2008, pp. 440–451.
- [36] A. Elsebai and F. Meziane, "Extracting person names from arabic newspapers," in *2011 International Conference on Innovations in Information Technology*. IEEE, 2011, pp. 87–89.
- [37] S. Abdallah, K. Shaalan, and M. Shoaib, "Integrating rule-based system with classification for arabic named entity recognition," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2012, pp. 311–322.
- [38] J. Maloney and M. Niv, "Tagarab: a fast, accurate arabic name recognizer using high-precision morphological analysis," in *Computational approaches to semitic languages*, 1998.
- [39] H. Traboulsi, "Arabic named entity extraction: A local grammar-based approach," in *2009 International Multiconference on Computer Science and Information Technology*. IEEE, 2009, pp. 139–143.
- [40] A. Elsebai, F. Meziane, F. Z. Belkredim, and U. H. B. Bouali, "A rule based persons names arabic extraction system."
- [41] T. Buckwalter, "Issues in arabic orthography and morphology analysis," in *proceedings of the workshop on computational approaches to Arabic script-based languages*, 2004, pp. 31–34.
- [42] F. Harrag, E. El-Qawasmeh, and A. M. Salman Al-Salman, "Extracting named entities from prophetic narration texts (hadith)," in *International Conference on Software Engineering and Computer Systems*. Springer, 2011, pp. 289–297.
- [43] M. Oudah and K. Shaalan, "A pipeline arabic named entity recognition using a hybrid approach," in *Proceedings of COLING 2012*, 2012, pp. 2159–2176.
- [44] Y. Benajiba, M. Diab, P. Rosso *et al.*, "Arabic named entity recognition: An svm-based approach," in *Proceedings of 2008 Arab international conference on information technology (ACIT)*. Association of Arab Universities Amman, Jordan, 2008, pp. 16–18.
- [45] 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*, vol. 8, 2008, pp. 143–153.
- [46] M. A. Bidhendi, B. Minaei-Bidgoli, and H. Jouzi, "Extracting person names from ancient islamic arabic texts," in *Proceedings of Language Resources and Evaluation for Religious Texts (LRE-Rel) Workshop Programme, Eight International Conference on Language Resources and Evaluation (LREC 2012)*, 2012, pp. 1–6.
- [47] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.
- [48] R. Koulali and A. Meziane, "A contribution to arabic named entity recognition," in *2012 Tenth International Conference on ICT and Knowledge Engineering*. IEEE, 2012, pp. 46–52.
- [49] A. Lotfy, C. Sabty, and S. Abdennadher, "Enhancing deep learning with embedded features for arabic named entity recognition."
- [50] M. Gridach, "Deep learning approach for arabic named entity recognition," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2016, pp. 439–451.
- [51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their



- compositionality," *Advances in neural information processing systems*, vol. 26, 2013.
- [52] Y. Zhao, Y. Shen, and J. Yao, "Recurrent neural network for text classification with hierarchical multiscale dense connections," in *IJCAI*, 2019, pp. 5450–5456.
- [53] J. C.-W. Lin, Y. Shao, Y. Djenouri, and U. Yun, "Asrnn: A recurrent neural network with an attention model for sequence labeling," *Knowledge-Based Systems*, vol. 212, p. 106548, 2021.
- [54] M. N. A. Ali, G. Tan, and A. Hussain, "Bidirectional recurrent neural network approach for arabic named entity recognition," *Future Internet*, vol. 10, no. 12, 2018. [Online]. Available: <https://www.mdpi.com/1999-5903/10/12/123>
- [55] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.
- [56] K. Shahina, P. Jyothsna, G. Prabha, B. Premjith, and K. Soman, "A sequential labelling approach for the named entity recognition in arabic language using deep learning algorithms," in *2019 International Conference on Data Science and Communication (IconDSC)*. IEEE, 2019, pp. 1–6.
- [57] M. Gridach, "Character-aware neural networks for Arabic named entity recognition for social media," in *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, Dec. 2016, pp. 23–32.
- [58] K. Darwish, "Named entity recognition using cross-lingual resources: Arabic as an example," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1558–1567.
- [59] D. Awad, C. Sabty, M. Elmahdy, and S. Abdennadher, "Arabic name entity recognition using deep learning," in *International Conference on Statistical Language and Speech Processing*. Springer, 2018, pp. 105–116.
- [60] M. N. A. Ali, G. Tan, and A. Hussain, "Boosting arabic named-entity recognition with multi-attention layer," *IEEE Access*, vol. 7, pp. 46 575–46 582, 2019.
- [61] C. Helwe and S. Elbassuoni, "Arabic named entity recognition via deep co-learning," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 197–215, 2019.
- [62] L. Liu, J. Shang, and J. Han, "Arabic named entity recognition: What works and what's next," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 60–67.
- [63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [64] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- [65] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *ArXiv*, vol. abs/2003.10555, 2020.
- [66] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.
- [67] A. Safaya, "Arabic-albert," Aug. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4718724>
- [68] A. Ghaddar, Y. Wu, S. Bagga, A. Rashid, K. Bibi, M. Rezagholizadeh, C. Xing, Y. Wang, D. Xinyu, Z. Wang *et al.*, "Revisiting pre-trained language models and their evaluation for arabic natural language understanding," *arXiv preprint arXiv:2205.10687*, 2022.
- [69] W. Antoun, F. Baly, and H. M. Hajj, "Aragpt2: Pre-trained transformer for arabic language generation," in *WANLP*, 2021.
- [70] E. M. B. Nagoudi, A. Elmadany, and M. Abdul-Mageed, "AraT5: Text-to-text transformers for Arabic language generation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, May 2022, pp. 628–647.
- [71] N. Alsaaran and M. AlRabiah, "Classical arabic named entity recognition using variant deep neural network architectures and bert," *IEEE Access*, vol. 9, pp. 91 537–91 547, 2021.
- [72] M. S. Al-Qurishi and R. Souissi, "Arabic named entity recognition using transformer-based-CRF model," in *Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*. Trento, Italy: Association for Computational Linguistics, 12–13 Nov. 2021, pp. 262–271. [Online]. Available: <https://aclanthology.org/2021.icnlsp-1.31>
- [73] C. Helwe, G. Dib, M. Shamas, and S. Elbassuoni, "A semi-supervised BERT approach for Arabic named entity recognition," in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, 2020, pp. 49–57.
- [74] M. El-Haj and R. Koulali, "Kalimat a multipurpose arabic corpus," in *Second workshop on Arabic corpus linguistics (WACL-2)*, 2013, pp. 22–25.
- [75] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "Arbert & marbert: Deep bidirectional transformers for arabic," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 7088–7105.
- [76] N. Boudjellal, H. Zhang, A. Khan, A. Ahmad, R. Naseem, J. Shang, and L. Dai, "Abioner: A bert-based model for arabic biomedical named-entity recognition," *Complex.*, vol. 2021, pp. 6 633 213:1–6 633 213:6, 2021.
- [77] 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.
- [78] W. Zaghouani, "Renar: A rule-based arabic named entity recognition system," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 11, no. 1, pp. 1–13, 2012.
- [79] A. Morsi and A. Rafea, "Studying the impact of various features on the performance of conditional random field-based arabic named entity recognition," in *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2013, pp. 1–5.
- [80] 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.
- [81] F. Dahan, A. Touir, and H. Mathkour, "First order hidden markov model for automatic arabic name entity recognition," *International Journal of Computer Applications*, vol. 123, no. 7, 2015.
- [82] F. B. Mesmia, K. Haddar, N. Friburger, and D. Maurel, "Casaner: Arabic named entity recognition tool," in *Intelligent natural language processing: Trends and applications*. Springer, 2018, pp. 173–198.
- [83] M. Khalifa and K. Shaalan, "Character convolutions for arabic named entity recognition with long short-term memory networks," *Computer Speech & Language*, vol. 58, pp. 335–346, 2019.
- [84] H. M. Elsherif, K. M. Alomari, A. Q. M. AlHamad, and K. Shaalan, "Arabic rule-based named entity recognition system using gate," in *MLDM (1)*, 2019, pp. 1–15.
- [85] N. Alsaaran and M. Alrabiah, "Arabic named entity recognition: A bert-bgru approach," *Comput. Mater. Contin.*, vol. 68, pp. 471–485, 2021.
- [86] A. Rom and K. Bar, "Supporting undotted arabic with pre-trained language models," in *Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, 2021, pp. 89–94.
- [87] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv e-prints*, pp. arXiv–1706, 2017.
- [88] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, pp. 30–43, 2018.
- [89] J. Worsham and J. Kalita, "Multi-task learning for natural language processing in the 2020s: where are we going?" *Pattern Recognition Letters*, vol. 136, pp. 120–126, 2020.
- [90] M. Jarrar, M. Khalilia, and S. Ghanem, "Wojood: Nested arabic named entity corpus and recognition using bert," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France, June, 2022.
- [91] S. Ahmed, R. van der Goot, M. Rehman, C. Kruse, Ö. Özsoy, A. Mehler, and G. Roig, "Tafsir dataset: A novel multi-task benchmark for named entity recognition and topic modeling in classical arabic literature," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 3753–3768.
- [92] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Larousilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.
- [93] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [94] 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. 37 736–37 745, 2020.
- [95] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung *et al.*, "Multilingual

- universal sentence encoder for semantic retrieval," *arXiv preprint arXiv:1907.04307*, 2019.
- [96] F. Alotaibi and M. Lee, "A hybrid approach to features representation for fine-grained arabic named entity recognition," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 984–995.
- [97] W. Lan, Y. Chen, W. Xu, and A. Ritter, "An empirical study of pre-trained transformers for arabic information extraction," *arXiv preprint arXiv:2004.14519*, 2020.
- [98] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified mrc framework for named entity recognition," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5849–5859.
- [99] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6470–6476.
- [100] J. Li, H. Fei, J. Liu, S. Wu, M. Zhang, C. Teng, D. Ji, and F. Li, "Unified named entity recognition as word-word relation classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10965–10973.
- [101] Y. Gu, X. Qu, Z. Wang, Y. Zheng, B. Huai, and N. J. Yuan, "Delving deep into regularity: A simple but effective method for Chinese named entity recognition," in *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1863–1873. [Online]. Available: <https://aclanthology.org/2022.findings-naacl.143>
- [102] Y. Shen, X. Ma, Z. Tan, S. Zhang, W. Wang, and W. Lu, "Locate and label: A two-stage identifier for nested named entity recognition," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 2782–2794.
- [103] C. Xia, C. Zhang, T. Yang, Y. Li, N. Du, X. Wu, W. Fan, F. Ma, and S. Y. Philip, "Multi-grained named entity recognition," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1430–1440.
- [104] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu, "Unified structure generation for universal information extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5755–5772.
- [105] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
- [106] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.
- [107] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various ner subtasks," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 5808–5822.
- [108] X. Chen, N. Zhang, L. Li, X. Xie, S. Deng, C. Tan, F. Huang, L. Si, and H. Chen, "Lightner: A lightweight generative framework with prompt-guided attention for low-resource ner," *arXiv preprint arXiv:2109.00720*, 2021.
- [109] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 753–757.
- [110] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.
- [111] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot filling models in natural language understanding," *ACM Computing Surveys (CSUR)*, 2021.
- [112] K. Mai, T.-H. Pham, M. T. Nguyen, T. D. Nguyen, D. Bollegala, R. Sasano, and S. Sekine, "An empirical study on fine-grained named entity recognition," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 711–722.
- [113] M. Peng, R. Ma, Q. Zhang, L. Zhao, M. Wei, C. Sun, and X.-J. Huang, "Toward recognizing more entity types in ner: an efficient implementation using only entity lexicons," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 678–688.
- [114] J. Wan, D. Ru, W. Zhang, and Y. Yu, "Nested named entity recognition with span-level graphs," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 892–903.
- [115] C. Lou, S. Yang, and K. Tu, "Nested named entity recognition as latent lexicalized constituency parsing," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6183–6198.
- [116] F. Li, Z. Lin, M. Zhang, and D. Ji, "A span-based model for joint overlapped and discontinuous named entity recognition," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4814–4828.
- [117] Y. Wang, B. Yu, H. Zhu, T. Liu, N. Yu, and L. Sun, "Discontinuous named entity recognition as maximal clique discovery," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 764–774.
- [118] X. Dai, S. Karimi, B. Hachey, and C. Paris, "An effective transition-based model for discontinuous ner," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5860–5870.
- [119] J. Liu, D. Ji, J. Li, D. Xie, C. Teng, L. Zhao, and F. Li, "Toe: A grid-tagging discontinuous ner model enhanced by embedding tag/word relations and more fine-grained tags," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 177–187, 2022.
- [120] X. Dai and H. Adel, "An analysis of simple data augmentation for named entity recognition," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 3861–3867.
- [121] B. Ding, L. Liu, L. Bing, C. Kruengkrai, T. H. Nguyen, S. Joty, L. Si, and C. Miao, "Daga: Data augmentation with a generation approach for low-resource tagging tasks," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6045–6057.
- [122] J. Liu, Y. Chen, and J. Xu, "Low-resource ner by data augmentation with prompting," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4252–4258.
- [123] L. Cui, Y. Wu, J. Liu, S. Yang, and Y. Zhang, "Template-based named entity recognition using bart," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 1835–1845.
- [124] D.-H. Lee, A. Kadakia, K. Tan, M. Agarwal, X. Feng, T. Shibuya, R. Mitani, T. Sekiya, J. Pujara, and X. Ren, "Good examples make a faster learner: Simple demonstration-based learning for low-resource ner," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 2687–2700.
- [125] Y. Yang, W. Chen, Z. Li, Z. He, and M. Zhang, "Distantly supervised ner with partial annotation learning and reinforcement learning," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2159–2169.
- [126] M. Peng, X. Xing, Q. Zhang, J. Fu, and X.-J. Huang, "Distantly supervised named entity recognition using positive-unlabeled learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2409–2419.
- [127] C. Liang, Y. Yu, H. Jiang, S. Er, R. Wang, T. Zhao, and C. Zhang, "Bond: Bert-assisted open-domain named entity recognition with distant supervision," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1054–1064.
- [128] Y. Meng, Y. Zhang, J. Huang, X. Wang, Y. Zhang, H. Ji, and J. Han, "Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 10367–10378.
- [129] X. Zhang, B. Yu, T. Liu, Z. Zhang, J. Sheng, X. Mengge, and H. Xu, "Improving distantly-supervised named entity recognition with self-collaborative denoising learning," in *Proceedings of the 2021*

- Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 10746–10757.
- [130] X. Qu, J. Zeng, D. Liu, Z. Wang, B. Huai, and P. Zhou, “Distantly-supervised named entity recognition with adaptive teacher learning and fine-grained student ensemble,” *arXiv preprint arXiv:2212.06522*, 2022.
  - [131] H. Jiang, D. Zhang, T. Cao, B. Yin, and T. Zhao, “Named entity recognition with small strongly labeled and large weakly labeled data,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 1775–1789.
  - [132] A. Jain, B. Paranjape, and Z. C. Lipton, “Entity projection via machine translation for cross-lingual ner,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1083–1092.
  - [133] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, “A survey of data augmentation approaches for nlp,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 968–988.
  - [134] P. Keung, Y. Lu, and V. Bhardwaj, “Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and ner,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1355–1360.
  - [135] W. Chen, H. Jiang, Q. Wu, B. Karlsson, and Y. Guan, “Advpicker: Effectively leveraging unlabeled data via adversarial discriminator for cross-lingual ner,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 743–753.
  - [136] G. Lample and A. Conneau, “Cross-lingual language model pretraining,” *arXiv preprint arXiv:1901.07291*, 2019.
  - [137] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Sidhant, A. Barua, and C. Raffel, “mt5: A massively multilingual pre-trained text-to-text transformer,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 483–498.
  - [138] A. Ebrahimi and K. Kann, “How to adapt your pretrained multilingual model to 1600 languages,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4555–4567.
  - [139] T. Pires, E. Schlinger, and D. Garrette, “How multilingual is multilingual bert?” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4996–5001.
  - [140] A. Sil and A. Yates, “Re-ranking for joint named-entity recognition and linking,” in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 2369–2374.
  - [141] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie, “Joint entity recognition and disambiguation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 879–888.
  - [142] P. H. Martins, Z. Marinho, and A. F. Martins, “Joint learning of named entity recognition and entity linking,” *ACL 2019*, p. 190, 2019.
  - [143] D. Zelenko, C. Aone, and A. Richardella, “Kernel methods for relation extraction,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1083–1106, 2003.
  - [144] Y. S. Chan and D. Roth, “Exploiting syntactico-semantic structures for relation extraction,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 551–560.
  - [145] M. Miwa and M. Bansal, “End-to-end relation extraction using lstms on sequences and tree structures,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1105–1116.
  - [146] J. Wang and W. Lu, “Two are better than one: Joint entity and relation extraction with table-sequence encoders,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1706–1721.