Comparing Open Arabic Named Entity Recognition Tools

Abdullah Aldumaykhi  
College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University  
Al-Kharj 11942, Saudi Arabia  
aldomaikhi@psau.edu.sa

Saad Ota  
College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University  
Al-Kharj 11942, Saudi Arabia  
441540018@psau.edu.sa

Abdulkareem Alsudais  
College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University  
Al-Kharj 11942, Saudi Arabia  
a.alsudais@psau.edu.sa

Abstract—The main objective of this paper is to compare and evaluate the performances of three open Arabic Named Entity Recognition (NER) tools: CAMeL, Hatmi, and Stanza. We collected a corpus consisting of 30 articles written in Modern Standard Arabic (MSA) and manually annotated all the entities of the person, organization, and location types at the article (document) level. Our results suggest a similarity between Stanza and Hatmi with the latter receiving the highest F1 score for the three entity types. However, CAMeL achieved the highest precision values for names of people and organizations. Following this, we implemented a “merge” method that combined the results from the three tools and a “vote” method that tagged named entities only when two of the three identified them as entities. Our results showed that merging achieved the highest overall F1 scores. Moreover, merging had the highest recall values while voting had the highest precision values for the three entity types. This indicates that merging is more suitable when recall is desired, while voting is optimal when precision is required. Finally, we collected a corpus of 21,635 articles related to COVID-19 and applied the merge and vote methods. Our analysis demonstrates the tradeoff between precision and recall for the two methods.

Keywords—Natural Language Processing, Named Entity Recognition, Platforms and Tools, Software and Systems Reuse and Reusability

I. INTRODUCTION

Named entity recognition (NER) is the task of identifying known entities such as people and organizations in text data. Methods for NER rely on predefined rules or lists, statistical information, machine learning techniques, and recently, large language models. NER can be used to enhance other natural language processing (NLP) and information retrieval applications such as question answering [1], [2] and semantics [3], [4]. Three of the most commonly used types of Named Entities (NEs) are Person (PER), Organization (ORG), and Location (LOC). Although the performance of computational models that automatically identify NEs in text data has improved in recent years, NER remains a research challenge. Moreover, research has traditionally focused on NER at the sentence level and limited research has focused on NER at the article or document levels [5]. For the Arabic language, several NER methods have been introduced over the years. A handful have ready-to-use tools or software that allow users to find NEs in documents written in Arabic. In this paper, our primary objective is to collect and annotate a new dataset that consists of Arabic documents and to compare the performance of three open Arabic NER tools as well as two of our own proposed approaches that integrate these three. Our sample is a leading Saudi newspaper, providing a specific focus on how these tools process names that are more common in the country.

Arabic is the official language in several countries in the Middle East and North Africa. It is the native language for over 300 million individuals and is considered one of the most morphologically diverse natural languages in the world. Some of the unique characteristics of Arabic, such as its lack of orthographic standards [6], [7], its diverse dialects [8], [9], and the limited existing NLP resources make Arabic NLP challenging. There are also problems that make Arabic NER particularly difficult such as Arabic’s “lack of capitalization” and “lack of uniformity in writing styles.” [10]. Despite this, several methods for Arabic NER currently exist.

Three tools that have recently been particularly successful for Arabic NER are Stanza, CAMeL, and hatmimoha/arabic-ner (Hatmi). Stanza is a natural language processing toolkit that supports several languages, including Arabic. CAMeL tools is an open-source toolkit that is focused on supporting tasks in Arabic NLP, including an Arabic NE tagger. Hatmi is a pre-trained BERT-based NER model for Arabic that relies on HuggingFace’s Transformers [11]. The three vary in their scope and the methods they use to identify Arabic NEs. To the best of our knowledge, no previous studies have examined these three tools and compared their performances when used on the same corpus at the article-level. This gap in the scholarship represents a main motivation for our work. More specifically, we are comparing the performance of these three tools on a dataset of Arabic news articles that were published in a leading Saudi newspaper. We chose to select a dataset that would provide a focus on this one country in order to investigate the ideal NER tool to use when the analyzed corpus included a higher number of NEs that are more common in Saudi. Additionally, we have tested “merging” and “voting” methods that utilize the results from these three tools. Finally, we have collected a subset of news articles containing the term “Covid-19” and discuss various observations about the use of these five options to identify NEs.

A study that focuses on comparing these three current NER tools can have several benefits. First, our comparative study identifies the method that performs best when used to identify NEs of the Person, Organization, and Location types. In order to furnish a dataset with many possible NEs, we collected a large...
dataset of Arabic news articles from a leading Saudi newspaper, then selected a random sample of 30 articles that we tagged manually. This new dataset represents a second benefit of our study: 30 articles with all the persons, organizations, and locations in each article manually identified and tagged at the article level. It was unclear whether combining the results obtained from the three models could generate results that were more accurate than using each model alone. Therefore, we tested a merge method that combines all the results obtained from the models independently and a vote method that only tags NEs when two of the three models recognize a NE. Finally, we collected a large dataset of news articles that contained mentions of COVID-19 and its related terms. Our analysis of this subset sheds light on how the five methods perform when used on a new dataset with many data and terms that are novel. In summary, the following are the key contributions of this paper:

- We collected a dataset of 21,635 articles written about COVID-19 and published in a Saudi newspaper and manually annotated a sample of 30 articles and tagged all NEs of the Person, Organization, and Location types at the document level. Our focus on this specific corpus aims to experiment with how the tools tested perform on this particular type of text corpora.
- We tested the performances of three open NER tools and quantified their results. This represents the first comparison of these three tools on the same corpus.
- We improved the results by testing two approaches: 1) a merging method that combines all the entities extracted from the three tools and 2) a voting method that only accepts a tagged NE when at least two of the methods identify it as a NE. This represents a major contribution of our work: Results for these two proposed methods outperform the results obtained when these models are used independently.

II. RELATED WORK

Arabic NER has been a research challenge for many years. Researchers have used various methods and techniques in an attempt to improve Arabic NER. Traditional approaches to the problem have relied on rule-based models [12], [13]. These approaches were followed by models that were based on machine learning [14], [15], as well as by hybrid approaches [16], [17]. Recently, several authors have experimented with deep learning techniques [18]–[20]. Researchers developing NER models have been using annotated corpora (i.e., corpora that include articles or sentences, with tagged entities in each article or sentence). There are several annotated corpora for Arabic NER. One popular dataset is AQMAR [21], which comprises 28 articles from Wikipedia with tagged NEs of the Person, Organization, Location, and Miscellaneous types. This corpus has often been used as a benchmark for new NER models [22], [23]. Another commonly references corpus is ANERcorp [24], which contains entities of the Person, Organization, and Location types. Both AQMAR and ANERcorp contain text written in Modern Standard Arabic. Some additional corpora have targeted specific domains. In this paper, we are focusing on a corpus written in MSA and consists of articles that were published in a Saudi newspaper.

III. METHODS

A. Dataset

In this study, we wanted to focus on articles published in a Saudi newspaper. To the best of our knowledge, no previous studies have focused on such a dataset. Due to the prevalence of certain names of people, organizations, and locations, we believed it was possible that one of the methods tested might perform better when used on such a subset. To collect the subset, we focused on Al-Riyadh newspaper, a leading Saudi newspaper. For each collected article, we captured relevant information such as the article’s ID, its publication date, title, and textual contents. We noticed that the publication dates were written in several formats. Therefore, we implemented a process to unify all publication dates. Since NER is sensitive to preprocessing steps that may remove stop words and punctuations, we did not perform any preprocessing for the textual contents of articles. In the end, we had a total of 242,724 IDs of articles that were published in the newspaper between 2017 and 2021. From these, we selected 1) a random sample of 30 that we manually annotated and 2) a subset of 21,635 articles that contained terms related to COVID-19.

B. Annotation

For the 30 randomly selected articles, we identified all NEs of the Person (PER), Organization (ORG), and Location (LOC) types. The number of articles in our annotated corpus is similar to the sizes of corpora used in other studies for Arabic NLP. To facilitate the annotation process, we used DataTurks annotation tool [25]. One important distinction in our annotation process is that we counted NEs at the article level. Put differently, if the same organization is mentioned twice in the same article, our annotation only counted it once. We decided to follow this annotation process because our subsequent analysis using the COVID-19 subset relies on counting NEs at the article level, a common counting strategy when analyzing large corpora. Such a strategy has the advantage of not overcalculating certain entities that appeared many times in the same article. Moreover, we believe this focus provides novel findings since our annotations does not focus on sentence-level annotation of NEs.

Two of the authors of this paper completed the annotation, which was then confirmed by the third author. To ensure agreement on what constitutes a NE of the PER, ORG, and LOC types, we selected an initial sample that contained only 10 articles. The two authors responsible for annotation completed the labeling process. Then, all three authors discussed the findings and confirmed a uniform understanding of the definitions (i.e. when to tag a word or phrase as a location). Following this step, we revisited the 30 articles, and the two annotators completed the labeling, which was then discussed and confirmed with the third author. We used this annotated sample of 30 articles to evaluate the various methods. Finally, our annotation resulted in identifying 54 NEs of the PER type, 64 of ORG, and 88 of LOC.

C. Approaches

Following the data collection and annotation process, we wanted to test the five approaches. In this section, we will describe each method. The first three are tools that have been provided by the scientific community. In our selection process, we focused on tools that are freely available and can be easily...
accessed. The rationale behind this selection condition is to only test tools and methods that are accessible and can be efficiently implemented by users. Moreover, we selected only tools that relied on methods that have been scientifically published.

a) Stanza: The first software library we tested was Stanza. It is a natural language processing toolkit created by the Stanford NLP Group [26]. The library is written in Python and includes support for several NLP tasks such as tokenization, sentiment analysis, and NER. Moreover, the library supports multiple languages such as English, Spanish, and Arabic. The Arabic NER model was evaluated on the AQMAR [21] corpus. The authors of Stanza indicated that their NER model relied on the utilization of “a forward and a backward character-level LSTM language model”, word embeddings, and a “standard one-layer Bi-LSTM sequence tagger with a conditional random field (CRF)-based decoder” [26]. Stanza is an extension of Stanford CoreNLP and is available as a Python package. In this study, we downloaded and implemented its Arabic NER model.

b) CAMeL: The second software library we used was CAMeL tools [27]. It includes a set of functions for a variety of NLP tasks such as sentiment analysis, Arabic dialect classification, and NER. One main difference between CAMeL and Stanza is that the former is specifically designed for and only supports Arabic. CAMeL is also available as a Python package. The software was developed by researchers from the CAMeL lab in New York University Abu Dhabi. The NER model in CAMeL was based on using “HugginFace’s Transformers” [11] to fine-tune AraBERT” [28]. According to the authors of CAMeL, their NER model was evaluated on the ANERcorp [29] and the overall F1 result was 83% while it was 90% for LOC, 63% for MISC, 73% for ORG, and 87% for PER. The model supports these four types of NEs.

c) Hatmi: Hatmi is a pre-trained BERT-based NER model for Arabic created by Mohamed Hatmi. Although there is no peer-reviewed paper that explains and evaluates the model, its author indicated that the model was based on Arabic-BERT-base (ArabicBERT) [31], which was pretrained on around 8.2 billion words. The model is available via HugginFace and supports the following entities (classes): person, organization, location, date, product, competition, prize, event, and disease. For the three models, we completed initial testing to ensure they met our requirements and that preliminary results would be promising when tested on our corpus.

d) Merge Method: The fourth approach we tested was to merge all the results from the previous three tools. When examining the initial results from a sample of articles, we noticed that the three tools did not always identify the same set of entities. This was the main motivation for testing the merge method. It works by combining the results from the models into one list. For example, for a sample paragraph, each model might identify the same organization. However, in some cases, we noticed that each model recognized a different organization. Thus, we inferred that combining all the results might improve performance. To implement this method, we wrote a function that combined all discovered NEs but captured the same entity only once. Table I includes a sample of manually annotated NEs from several sentences and the results reached when we used the “merge” method. We used “NaN” when one method did not identify any NEs of the specified type. For example, the first row shows only one entity of the “PER” type and all three were able to recognize it. In the third row, we only labeled four entities, but combining all those discovered from the three methods resulted in a list of six entities. Two of these six were inaccurately tagged as NEs of the Location type. Thus, this was an indicator that this method may not actually improve results.

e) Vote Method: One issue with merging all the results from the three models is that any mistakenly identified entity by any of the three will be included in the results for the merge method. Therefore, we wanted to also test a voting method. This method identifies a NE only when two or more of the three tools identify it as such. Since we observed that many mistakes or false positives only occurred in one of the three models, we tested this method to quantify whether it reduced the number of these false positives. Additionally, we wanted to examine if implementing this method would eliminate what could be the merge method’s strongest advantage: its ability to capture all the existing NEs. Table I includes the same three examples of the manually labeled entities as well as the results obtained using the vote method. As the results show, in row 2, this method eliminated the correct organization which was only identified by CAMeL. However, in the final row, the two NEs inaccurately captured by the merge method were not identified as locations by the vote method. These initial results provided justifications for the experimentation on and quantification of the performances of the five approaches. The table also includes translations for the Arabic tables in the first two rows.

| Type       | Annotated Entities | CAMeL | Stanza | Hatmi | Merge | Vote |
|------------|--------------------|-------|--------|-------|-------|------|
| PER        | [عمر الفريح]        | Omar Alfurīh | [عمر الفريح] | Omar Alfurīh | [عمر الفريح] | Omar Alfurīh |
| ORG        | Facebook           | مصطفى محمد محرز | [عمر الفريح] | Omar Alfurīh | NaN | Facebook |
| LOC        | [إسبيرونا]         | ظبيح أحمد الرازي | [عمر الفريح] | Omar Alfurīh | [عمر الفريح] | Omar Alfurīh |

TABLE I. EXAMPLES OF RESULTS FOR THE FIVE APPROACHES
D. Evaluation

To evaluate the performances of the five approaches, we used three standard metrics: precision, recall, and F1 scores. The three are commonly used to evaluate models that identify NEs in text [32], [33]. Calculating the three relied on identifying True Positive (TP), False Positive (FP), and False Negative (FN) cases. Using our annotated lists of NEs from the 30 articles, we measured how each method performed based on its TP, FP, and FN numbers. We defined TP instances as those where we annotated a NE that the model also identified and tagged as such. For FP, this took place when the model identified an entity that we did not annotate as an entity. Finally, FN were instances when we annotated an entity that the method did not identify. It is important to note here that for each method, we calculated these values for the three types of entities (PER, ORG, and LOC) and then for the three combined (four total). Once these values were calculated, we used them to obtain the results for the precision, recall, and F1 scores.

E. Analyzing a “COVID-19” Corpus

The final step in our work was to analyze the performances of the methods when applied on a COVID-19 corpus. The objective here was to examine how these methods performed when applied to a collection of documents that included discussions of a developing story that was likely not seen when training these methods. To collect the COVID-19 corpus, we amassed 21,635 articles published in the same newspaper. All of the methods when applied to a collection of documents that included discussions of a developing story that was likely not seen when training these methods. To collect the COVID-19 corpus, we amassed 21,635 articles published in the same newspaper. All of the articles had one or more of the following words: “coronavirus,” “corona,” “covid” as well as “corona” and “covid” when written in Arabic. The sixth and final word was the Arabic word for “pandemic” which is "جائحة".

IV. RESULTS

A. Results for Stanza, CAMeL, and Hatmi

The results for the three methods varied across the three entity types. The detailed results are in Table I. CAMeL had the highest precision value for PER with 0.83 compared to 0.64 and 0.71 for Stanza and Hatmi, respectively. Results show CAMeL receiving high precision and low recall values for all three entity types. However, precision values were still higher for Stanza and Hatmi when results for the three entity types were combined. Precision and recall results for Stanza and Hatmi were similar, and Hatmi had slightly higher values for precision while Stanza had higher recall values. Additionally, when all the entity types were combined, F1 scores showed that Hatmi had the highest performances compared to the other two methods. This suggests that when given the option to select only one of the three to identify NEs in similar corpora, Hatmi should be selected. Despite this, the F1 score result for Stanza was similar. Thus, existing frameworks that rely on that library may not benefit from switching to Hatmi. Finally, F1 scores for CAMeL were significantly lower, presumably due to CAMeL’s lower recall values. Still, the tool scored highest on some of the metrics. For example, it had the highest precision value when identifying names of people. Ultimately, the varying performances and clear strengths and weaknesses of each tool prompted us to consider combining the results from the three. Table I compares the five approaches. The highest performances for each measure and NE type in the table are displayed in bold.

### Table I. Summary of the Results for the Five Approaches

| NE Type | Method | Precision | Recall | F1 Scores |
|---------|--------|-----------|--------|-----------|
| PER     | Stanza | 0.648     | 0.648  | 0.648     |
|         | CAMeL | 0.833     | 0.092  | 0.166     |
|         | Hatmi | 0.717     | 0.611  | 0.66      |
|         | Merge | 0.578     | 0.814  | 0.676     |
|         | Vote  | 0.964     | 0.5    | 0.658     |
| ORG     | Stanza | 0.277     | 0.156  | 0.2       |
|         | CAMeL | 0.318     | 0.109  | 0.162     |
|         | Hatmi | 0.264     | 0.281  | 0.272     |
|         | Merge | 0.264     | 0.484  | 0.342     |
|         | Vote  | 0.444     | 0.062  | 0.109     |
| LOC     | Stanza | 0.396     | 0.454  | 0.423     |
|         | CAMeL | 0.421     | 0.181  | 0.253     |
|         | Hatmi | 0.5       | 0.375  | 0.428     |
|         | Merge | 0.358     | 0.545  | 0.432     |
|         | Vote  | 0.58      | 0.329  | 0.420     |
| ALL     | Stanza | 0.445     | 0.412  | 0.428     |
|         | CAMeL | 0.424     | 0.135  | 0.205     |
|         | Hatmi | 0.466     | 0.407  | 0.435     |
|         | Merge | 0.376     | 0.597  | 0.461     |
|         | Vote  | 0.689     | 0.291  | 0.409     |

B. Results for the Merge and Vote Methods

Following our individual evaluation of each tool, we combined all the results from the three and used the identified NEs to evaluate the results for the two. Results for the merge and vote methods for the different entity types show a similar pattern. For the three and including the total (all), the merge method scored highest for recall and F1 scores, while the vote method performed best for precision. This result represents another major finding of this work: Using the merge or vote methods will yield better performances than using Stanza, CAMeL, or Hatmi individually. The results for the merge and vote methods are also in Table I. Overall, voting between the three models generated higher precision values. This is perhaps because voting reduced false positive cases that only one of the three identified as a NE. These results indicate that when precision is most desirable, voting does produce the best values while merging has the highest overall scores for F1 and recall. Therefore, selecting either of these two methods might depend on the particular use case and application. The results also highlight some of the differences between the approaches. For example, the recall value for voting when used on ORG was very low. This suggests that there were few true positive instances of ORG that two of the three models agreed on.

C. Results for the COVID-19 Corpus

After determining which method worked best, we wanted to apply the merge and vote methods to a new, large corpus, specifically the 21,635 articles from a Saudi newspaper that contained reference to the COVID-19 pandemic. The results illustrated the tradeoff between precision and recall for the merge and vote methods. While the merge methods identified a significantly larger number of NEs, many of the names that appeared to be the most frequent were incorrectly identified as NEs. For example, “covid” and “corona” were two of the most frequent NEs of the person type found using the merge method.
Interestingly, “corona” was also the most frequent NE for ORG and LOC. Additionally, four of the most frequent LOCs were COVID-19 related terms that included “corona,” “coronavirus,” “covid,” and “novel corona.” Nevertheless, the remaining entities on the lists of the ten most frequent NEs were identified as 1) accurate names of entities that were 2) involved in or affected by the pandemic. These included organizations such as the World Health Organization, the Saudi Ministry of Health, and the United Nations. Our approach to improve the results obtained using the merge method is to use expert judgements. Put differently, an expert can examine the lists of most frequent NEs and remove those that are not correct names of entities.

For the vote method, we observed that none of the ten most frequent PERs, ORGs, or LOCs (30 total) were inaccurate. This represented an additional demonstration of the value of the high precision results for the vote method. Moreover, none of the “COVID-19” related terms appeared as a top PER, ORG, or LOC. Additionally, we observed that the city “Riyadh” was identified as both a top organization and a location using the merge method, while voting only listed it as a top location. Still, voting does suffer from low recall, which results in a failure to identify many of the NEs that it should capture. For example, the voting method identified 545 articles with the ORG “World Health Organization” while the merge method identified 875.

Table III includes the ten most frequent named entities of the ORG type that were identified using the merge and vote methods. The English terms in the table are the direct translations of the Arabic NEs.

The results also highlighted some of the difficulties associated with Arabic named entity recognition in general. First, we observed that the names of non-Arabic individuals and organizations were often written using two or more spellings. For example, the names “Donald Trump” appeared twice (two different spellings) in the list of the 10 most frequent PERs. Second, because some prepositions in Arabic join the words they precede, most frequent lists included the names of organizations twice: once with a preposition and once without. For example, the “World Health Organization” was once listed as that (“الصحة العالمية”) and once with the preposition “to” (“المنظمة العالمية”). Finally, we noticed instances where the same name was written either with or without the inclusion of a middle name. Similarly, we observed that sometimes the names of individuals were written with the Arabic word “son of” (“ابن”) before the middle (father) name. In all of these examples, it appeared as though the two names were different when they in fact belonged to the same real-world entity. Ultimately, these grammatical and spelling conventions in Arabic represent some of the several challenges that face Arabic named entity recognition. Some of which might be specific or more common in newspaper articles from Saudi Arabia.

V. CONCLUSION

In this paper, we annotated a collection of 30 articles and tagged all the named entities of the PER, ORG, and LOC types. We then tested three popular Arabic named entity recognition tools and evaluated their results. Moreover, we experimented with combining the results from the three tools using merge and vote methods. Our findings indicated that the vote method yielded the highest precision values while the merge method generated the highest recall and F1 score values. These are the key findings of our work. Following this, we applied the merge and vote methods on a corpus of articles about COVID-19, discussed several observations about the named entities discovered, and examined the list of most frequent NEs identified using the two approaches.

Despite the utility of this work, it also has several potential limitations. First, our selection criteria for tools focused on those that have accessible software implementations and avoided others that were only discussed in papers without having a tool that could be downloaded and utilized. Including these other papers could have resulted in the selection of models with better performances than those we selected. Second, our annotated corpus included articles written in a Saudi newspaper, and thus, our results may not be generalizable to all documents written in Modern Standard Arabic. For example, many of the NEs of the PER type were Saudi individuals. It is possible that one of the models does not capture such names. While we believe that these limitations are negligible, they still represent opportunities for future research which could focus on additional models and investigate whether results for the merge and vote methods change. Our approach could also be tested on other corpora. In the end, we believe our work presents several key contributions that help provide a better understanding of the issue of Arabic NER at the article-level and the possibilities associated with testing a variety of its tools and models.

### Table III. The Most Frequent Organizations Found Using the Merge and Vote Methods

| # | Merge Method Organization | Vote Method Organization | # | Merge Method Organization | Vote Method Organization |
|---|---------------------------|--------------------------|---|---------------------------|--------------------------|
| 1 | Corona (كرونا) | World Health Organization (منظمة الصحة العالمية) | 10 | to the United Nations (ال الأمم المتحدة) | British Government (الحكومة البريطانية) |
| 2 | Ministry of Health (وزارة الصحة) | to the World Health Organization (منظمة الصحة العالمية) | 9 | European Union (اتحاد الأوروبي) | Omani Ministry of Health (وزارة الصحة العمانية) |
| 3 | Kingdom (الملكة) | Riyadh (الرياض) | 8 | Kuwait Ministry of Health (وزارة الصحة الكويتية) | European Union (الاتحاد الأوروبي) |
| 4 | World Health Organization (منظمة الصحة العالمية) | 875 | Ministry of Health (وزارة الصحة) | 8 | to the United Nations (ال الأمم المتحدة) | 293 |
| 5 | Council of Ministers (مجلس الوزراء) | Real Madrid ( ريال مدريد) | 7 | United Nations (ال الأمم المتحدة) | World Health Organization (منظمة الصحة العالمية) |
| 6 | Kingdom of Saudi Arabia (الملكة العربية السعودية) | world health organization (الصحة العالمية) | 6 | Saudi Ministry of Health (وزارة الصحة السعودية) | 142 |
| 7 | Riyadh (الرياض) | Kingdom of Saudi Arabia (الملكة العربية السعودية) | 5 | Kingdom of Saudi Arabia (الملكة العربية السعودية) | 549 |
| 8 | Twitter (توتر) | Kuwait (كويت) | 4 | University of Oxford (الجامعة الأوروبية) | Omani Ministry of Health (وزارة الصحة العمانية) |
| 9 | European Union (اتحاد الأوروبي) | 394 | United Kingdom (المملكة المتحدة) | 4 | World Health Organization (منظمة الصحة العالمية) | 77 |
| 10 | to the United Nations (ال الأمم المتحدة) | British Government (الحكومة البريطانية) | 3 | World Health Organization (منظمة الصحة العالمية) | 148 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
REFERENCES

[1] T. H. Phan V and P. Do, “NERQUES: combining named entity recognition and sequence to sequence to automatically generating Vietnamese questions,” Neural Comput Appl, 2021, doi: 10.1007/s00521-021-06477-7.

[2] T. H. Alwaneeen, A. M. Azmi, H. A. Aboalsamh, E. Camibia, and A. Hussain, “Arabic question answering system: a survey,” Artif Intell Rev, 2021, doi: 10.1007/s10462-021-10031-1.

[3] A. Alsudaiai and H. Tchalian, “Clustering prominent named entities in topic-specific text corpus,” in 25th Americas Conference on Information Systems, AMCIS 2019, 2019.

[4] P. Khare, G. Burel, D. Maynard, and H. Alani, “Cross-lingual classification of crisis data,” in International Semantic Web Conference, 2018, pp. 617–633.

[5] L. Wang, S. Li, Q. Yan, and G. Zhou, “Domain-specific named entity recognition with document-level optimization,” ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol. 17, no. 4, pp. 1–15, 2018.

[6] N. Habash et al., “Unified Guidelines and Resources for Arabic Dialect Orthography,” in Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[7] S. Harrat, K. Meltof, and K. Smaili, “Machine translation for Arabic dialects (survey),” Inf Process Manag, vol. 56, no. 2, pp. 262–273, 2019, doi: 10.1016/j.ipm.2017.08.003.

[8] K. Darwish et al., “A Panoramic Survey of Natural Language Processing in the Arab World,” Commun ACM, vol. 64, no. 4, pp. 72–81, Mar. 2021, doi: 10.1145/3447735.

[9] A. Alsudaiai, W. Aloatibi, and F. Alomary, “Similarities between Arabic dialects: Investigating geographical proximity,” Inf Process Manag, vol. 59, no. 1, p. 102770, 2022, doi: https://doi.org/10.1016/j.ipm.2021.102770.

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

[11] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.

[12] W. Zaghouni, “RENAR: A Rule-Based Arabic Named Entity Recognition System,” ACM Transactions on Asian Language Information Processing, vol. 11, no. 1, Mar. 2012, doi: 10.1145/2090176.2090178.

[13] M. Oudah and K. Shaalan, “NERA 2.0: Improving coverage and performance of rule-based named entity recognition for Arabic,” Nat Lang Eng, vol. 23, no. 3, pp. 441–472, 2016, doi: 10.1017/S1353324916000097.

[14] 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), Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1558–1567.

[15] 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, doi: 10.18653/v1/N16-3003.

[16] M. Oudah and K. Shaalan, “A Pipeline Arabic Named Entity Recognition using a Hybrid Approach,” in Proceedings of COLING’2012, Mumbai, India: ThE COLING 2012 Organizing Committee, Dec. 2012, pp. 2159–2176.