Egyptian Informatics Journal 22 (2021) 295–302

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

The impact of using different annotation schemes on named entity recognition

Nasser Alshammari *, Saad Alanazi

College of Computer and Information Sciences, Department of Computer Science, Jouf University, Saudi Arabia

ARTICLE INFO

Article history:
Received 10 February 2020
Revised 30 September 2020
Accepted 30 October 2020
Available online 19 November 2020

Keywords:
Annotation schemes
Named entity recognition
Natural language processing
Segment representation

ABSTRACT

Named entity recognition (NER) is a subfield of information extraction, which aims to detect and classify predefined named entities (e.g., people, locations, organizations, etc.) in a body of text. In the literature, many researchers have studied the application of different machine learning models and features to NER. However, few research efforts have been devoted to studying annotation schemes used to label multi-token named entities. In this research, we studied seven annotation schemes (IO, IOB, IOE, IOBES, BI, IE, and BIES) and their impact on the task of NER using five different classifiers. Our experiment was conducted on an in–house dataset that consists of 27 medical Arabic articles with more than 62,000 tokens. The IO annotation scheme outperformed other schemes with an F-measure score of 84.44%. The closest competitor is the BIES scheme, which scored 72.78%. The rest of the schemes' scores ranged from 60.38% to 69.18%. Although the IO scheme achieved the best results, comparing it to the other schemes is not reasonable because it cannot identify consecutive entities, which the other schemes can do. Therefore, we also investigated the ability of recognizing consecutive entities and provided an analysis of the running-time complexity.

1. Introduction

Named entity recognition (NER) is a common task in the field of natural language processing (NLP). An NER system is designed to search for specific expressions and words based on their meaning, such as the names of people, locations, and organizations [1]. Correctly identifying and categorizing named entities can often be the key to deciphering the meaning of the analyzed text. Although there are several research efforts concerning NER, especially focusing on different machine learning methods, few research efforts have been devoted to studying the annotation schemes used to label multi-token named entities in English and other languages. This could be an issue, as named entities rarely consist of a single token, which means that the results can be inaccurate if some tokens of the named entity are not correctly identified. When it comes to the Arabic language specifically, the literature, up to the authors' knowledge, lacks any sufficient study that explore the impact and effects of using multi-annotation schemes on Arabic NER tasks. To redeem this knowledge gap, our research aims to study the impact of using different annotation schemes on the task of Arabic NER. To achieve this aim, we set several objectives which are: building a multi-scheme representative dataset, conducting a thorough experiment using five machine learning classifiers, and gathering and interpreting the outcomes of the experiment.

Several annotation schemes have been used in the literature. However, choosing the ideal annotation scheme is a complex problem [2]. In this study, seven annotation schemes are explored in terms of their influence on the task of Arabic NER. These schemes are the following:

- **IO:** is the simplest scheme that can be applied to this task. In this scheme, each token from the dataset is assigned one of two tags: an inside tag (I) and an outside tag (O). The I tag is for named entities, whereas the O tag is for normal words. This scheme has a limitation, as it cannot correctly encode consecutive entities of the same type.
- **IOB:** This scheme is also referred to in the literature as BIO and has been adopted by the Conference on Computational Natural Language Learning (CoNLL) [1]. It assigns a tag to each word in

* Corresponding author.
E-mail addresses: nashamri@ju.edu.sa (N. Alshammari), sanazi@ju.edu.sa (S. Alanazi).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

https://doi.org/10.1016/j.eij.2020.10.004

the text, determining whether it is the beginning (B) of a known named entity, inside (I) it, or outside (O) of any known named entities.

- **IOE:** This scheme works nearly identically to IOB, but it indicates the end of the entity (E tag) instead of its beginning.
- **IOBES:** An alternative to the IOB scheme is IOBES, which increases the amount of information related to the boundaries of named entities. In addition to tagging words at the beginning (B), inside (I), end (E), and outside (O) of a named entity. It also labels single-token entities with the tag S.
- **BI:** This scheme tags entities in a similar method to IOB. Additionally, it labels the beginning of non-entity words with the tag B-O and the rest as I-O.
- **IE:** This scheme works exactly like IOE with the distinction that it labels the end of non-entity words with the tag E-O and the rest as I-O.
- **BIES:** This scheme encodes the entities similar to IOBES. In addition, it also encodes the non-entity words using the same method. It uses B-O to tag the beginning of non-entity words, I-O to tag the inside of non-entity words, and S-O for single non-entity tokens that exist between two entities.

The rest of this paper is organized as follows. Section 2 surveys the related work in the literature. Section 3 illustrates the proposed methodology to measure the impact of using different annotation schemes on Arabic NER task. Section 4 presents the results and a discussion of the findings. Section 5 concludes this paper.

2. Related work

The Arabic NER research field saw a steady growth in recent years. This growth can be attributed to recent developments in artificial intelligence techniques and to the availability of language resources. The majority of these recent research efforts focused on applying deep learning techniques to Arabic NER [3–5]. Other efforts focused on developing new Arabic language tools such as CAMEL [6] and CasANER [7]. After surveying the literature in this field, it was apparent that there is a lack of research efforts that deals with studying the issue of annotation schemes in Arabic NER tasks. However, there are few research works that explored the impact of annotation schemes on NER for other languages which will be presented in this section.

Cho et al. [8] presented a method for feature generation, which expands the feature space to include multiple annotation schemes. This allows the assigned model to use the most discriminating features of a complex annotation scheme but also to avoid the issue of data-sparseness by incorporating the features of simple annotation schemes. This method incorporates several annotation schemes in the place of feature functions to support conditional random fields (CRF). This means that this well-established procedure can be applied to training. This study shows that it is possible to increase the tagging speed of a model by applying multiple annotation schemes so that it matches the speed of a more complex annotation scheme. This method has also been evaluated against two NER tasks previously: BioCreative 2 gene mention recognition [9], and CoNLL NER shared task [1].

Tkachenko et al. [10] used two of the more frequently used tagging schemes, BIO and the more complex BILOU, for NER in Estonian. The results discussed in this paper show that BILOU with its more detailed tagging system, performed slightly better than BIO; however, the F-measure values only increased very slightly, from 86.7% to 87%.

Konkol and Konopík [2] inspected the effect that different segment representations have on the task of NER. The authors claimed that the choice of segment representations is commonly an arbitrary decision. Therefore, the authors conducted several experiments to study different segment representations. All the segment representations were tested using CRF and maximum entropy (ME), which are both fairly popular machine learning algorithms. The tests were applied in four different languages: Czech, Dutch, English, and Spanish. In addition, BILOU performed the worst for English when using CRF, whereas the IOE-1 and IOE-2 versions seemed to perform the best across all languages and methods. This suggests that the choice of using simpler approaches, such as BIO or IOE, or more complex ones like BILOU, is not as clear-cut as it might appear. This research highlights that the best approach varies according to all the factors.

Malik and Sarwar [11] proposed a tagging scheme (BIL2) that is similar to IOBES. The only difference is that the single-token entities are labeled with the tag L. This scheme was proposed as potentially effective for subject object verb (SOV) languages that contain postpositioning. Using Urdu as their case study and applying the hidden Markov model (HMM) and CRF, they compared the results of various tagging schemes: IO, BIO2, BILOU, and BIL2. The results showed that BIL2 achieved the highest F-measure in their experiment.

Mozharova and Loukachevitch [12] applied two different labeling schemes to Russian text: IO and BIO, using a CRF classifier. The BIO scheme performed well against IO. This might be due to the structure of the Russian language because named entities are usually located next to each other. Their results highlight the weaknesses of IO annotation scheme in the case of consecutive named entities. As we can see, different languages may benefit from specific annotation schemes. This is natural, as languages' grammatical structures can affect the way named entities appear. This means that the way we isolate and categorize named entities in each language is different.

Table 1 shows a comparison between previous studies and our current study. Although Arabic NER is a field that is currently experiencing continual growth, as far as the authors have found, no similar work has been done on the effect of different annotation schemes in Arabic NER. Currently, many sophisticated features and techniques are being applied to Arabic NER, but the effectiveness of annotation schemes has not yet been investigated. With Arabic displaying such unique morphological challenges, it is essential that we establish which of the current NER annotation schemes is the most effective and to explore this research field.

3. Methodology

The framework used in this study consists of three main stages. The first stage is concerned with dataset preparation and includes data collection, pre-processing and annotation. The second stage is the experimental study which is accomplished in two steps, feature engineering and the classification. The final stage is concerned with the evaluation of the models results using several evaluation metrics. The framework is illustrated using Fig. 1.

Table 1
Comparisons of similar previous works with ours.

Research	Year	Schemes	Language(s)
Cho et al. [8]	2013	7	English
Tkachenko et al. [10]	2013	2	Estonian
Konkol and Konopik [2]	2015	10	English, Spanish, Dutch, Czech
Malik and Sarwar [11]	2016	4	Urdu
Mozharova and Loukachevitch [12]	2016	2	Russian
Our work	2020	7	Arabic

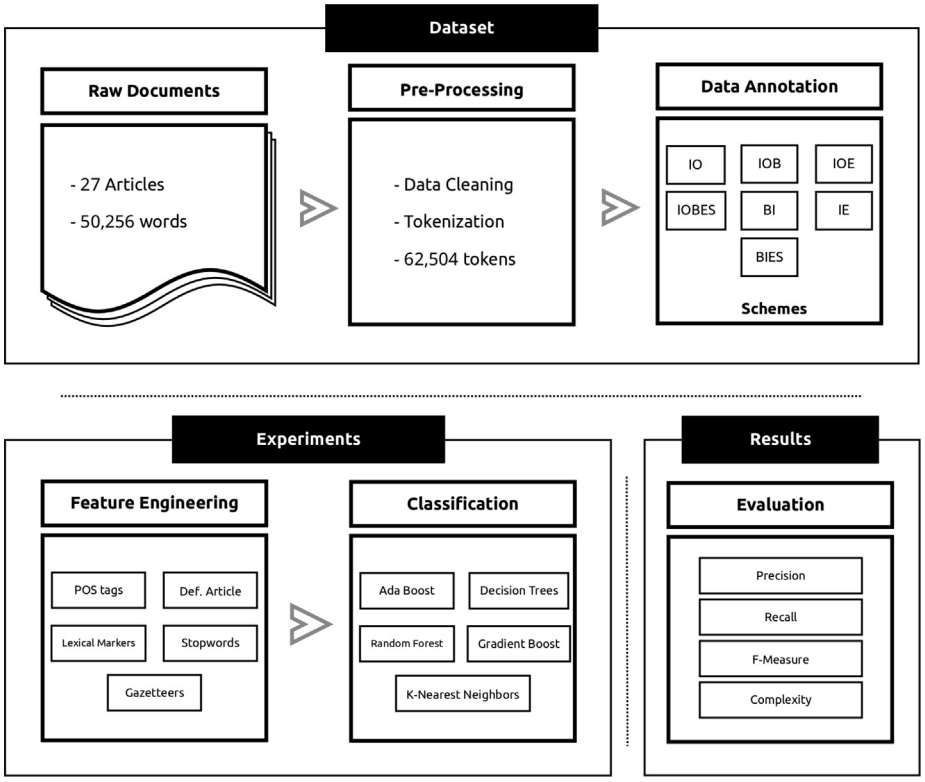

Fig. 1. The system framework.

3.1. Dataset

Our dataset was collected from the King Abdullah Bin Abdulaziz Arabic Health Encyclopedia (KAAHE) website. The encyclopedia was the result of the collaboration between King Saud Bin Abdulaziz University for Health Sciences, the Saudi Association for Health Informatics, the National Guard Health Affairs, the Health on the Net Foundation, the World Health Organization, and the National Health Services (NHS) in the UK [13].

The data were extracted from 27 articles where a total of 50,256 words were acquired. A preprocessing step was carried out by tokenizing the words where all prefixes (e.g., conjunctions, prepositions, etc.) were segmented off. Fig. 2 shows an example sentence before and after the preprocessing step. The final number of tokens was 62,504, and the number of entities was 1,278 [14].

The data were linguistically analyzed based on the frequency, collocation, and concordance, which led to identifying the features shown in Table 2. These features were calculated for each token in our dataset. Then, the data were manually annotated by two independent annotators according to the IO and IOB schemes. To ensure the reliability of the annotation process, the inter-annotator agreement (Cohen's kappa metric [15]) was calculated and its score was 95.14%, indicating a high agreement between the two annotators.

```
def genIOE(input_dataset, TARGET_LABEL):
   # creating a new copy of the input dataset
   output_dataset = input_dataset.copy()
   # iterating over the output dataset's rows
   for i, r in output_dataset.iterrows():
      # mapping between the two schemes token by token
      if i == 0:
         if r[TARGET_LABEL] == "I":
            output_dataset.at[i, TARGET_LABEL] = "I"
         else:
            output_dataset.at[i, TARGET_LABEL] = "O"
      if i == len(input_dataset) - 1:
         continue
      if r[TARGET_LABEL] == "I" and input_dataset.iloc[i + 1][
          TARGET_LABEL] == "O":
         output_dataset.at[i, TARGET_LABEL] = "E"
   return output_dataset
```

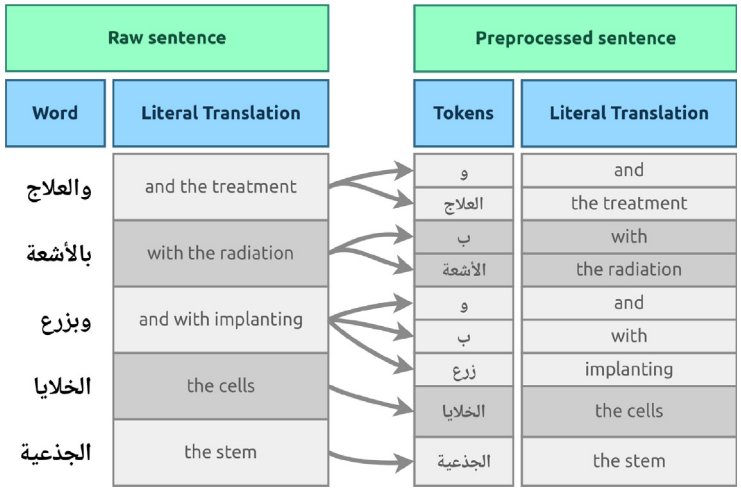

Fig. 2. Illustration of the preprocessing step.

Table 2
The list of features used to train the classifiers

Feature	Description
POS tags	A categorical feature for part-of-speech tags (noun, adj., etc.)
Lexical markers	A boolean feature that indicates the presence of the word in a lexical trigger dictionary
Stop words	A boolean feature that indicates the existence of the word in the stop-word list
Gazetteers	A boolean feature that indicates the existence of the word in the list of most common entities
Definiteness	A boolean feature that indicates the presence of the definite article at the beginning of the word

The annotation process for the rest of the annotation schemes (IOE, IOBES, BI, IE, and BIES) was completed using a script that generated the datasets. Table 3 shows a sentence annotated using each of the previously described annotation schemes in Section 1.

3.2. Experiment

To draw reliable information and increase the confidence in our experiment, we used cross-validation with five folds for each classifier. These classifiers are the following:

- Ada Boost [16]
- Decision Tree [17]
- K-nearest neighbors (KNN) [18]
- Random Forest [19]
- Gradient Boost [20]

The aforementioned classifiers were selected due to their popularity and to establish a baseline that shows the impact (or lack thereof) of the annotation scheme when performing NER tasks.

The performance of these classifiers is measured by well-known metrics: precision, recall, and F-measure. Although these evaluation metrics are heavily used, in the NER task, specifically, three standards should be considered when applying these evaluation metrics. These standards are message understanding conference (MUC) [21], computational natural language learning (CoNLL) [1], and automatic content extraction (ACE) [22]. They deal with the issue of boundary errors in multi-token entities.

The MUC [21] gives a partial score when the model correctly predicts parts of the multi-token entities. In contrast, CoNLL [1] is an aggressive metric that does not assign a partial score. An exact match of the entities as a whole and a correct classification must be identified to earn credit. This method of scoring is popular because of its simplicity in calculating and analyzing results. The third standard is ACE [22], which considers other factors, such as mention detection and co-reference resolution.

In this paper, we adopted CoNLL as an evaluation metric due to its abundant usage in the Arabic NER tasks in the literature. According to [23], this evaluation metric is heavily used for Arabic NER due to its simplicity in calculating and analyzing the results. Listing 2 shows a code snippet that details how CoNLL metric was adopted in this work.

Table 3
A sample sentence annotated using the seven annotation schemes

Word	Literal Translation	IO	IOB	IOE	IOBES	BI	IE	BIES
تعتبر	Considered	O	O	O	O	BO	EO	SO
اللوكيميا	Leukemia	I	B	E	S	B	E	S
)	)	O	O	O	O	BO	EO	SO
سرطان	Cancer	I	B	I	B	B	I	B
خلايا	Cells	I	I	I	I	I	I	I
الدم	Blood	I	I	I	I	I	I	I
البيضاء	White	I	I	E	E	I	E	E
(	(	O	O	O	O	BO	IO	BO
من	one of	O	O	O	O	IO	IO	IO
أكثر	the most	O	O	O	O	IO	IO	IO
الأنواع	kinds	O	O	O	O	IO	IO	IO
انتشاراً	common	O	O	O	O	IO	EO	IO

```
# CoNLL function returns the number of ture positives, false positives,
    and false negatives. CoNLL takes the ground truth (y_test) and the
    system prediction (y_pred).

def CoNLL(y_test, y_pred):
   # get_bounds function returns a list that determins the boundaries of
       each entity
   test_ents = get_bounds(y_test)
   pred_ents = get_bounds(y_pred)
   tp = fp = 0
   # looping over the predicted tokens of an entity, i = index, x = token
   for i, x in enumerate(pred_ents):
      if x in test_ents:
         if x == test_ents[test_ents.index(x)]:
            tp += 1 # only when an exact match occurs
      else:
         fp += 1 # otherwise, add one to false positives
   # calculating false negatives by substracting the number of entities
       in the ground truth list from the true positives
   fn = len(test_ents) - tp
   return (tp, fp, fn)
```

4. Results and discussion

The results of the experiment are presented in terms of the F-measure, precision, and recall in Tables 4–6 respectively. A visualization of the reported F-measure results is shown in Fig. 3. It is clear that the IO annotation scheme outperformed the other schemes in terms of F-measure, as it achieved a score of 84.44% for all classifiers. The closest competitor is the BIES scheme, which scored 72.78%. The rest of the schemes' scores ranged from 60.38% to 69.18%.

Besides the importance of the F-measure as the harmonic mean of the recall and precision, reporting the recall and precision

Table 4
The reported F-measure score for each classifier and annotation scheme

	IO(%)	IOB(%)	IOE(%)	IOBES(%)	BI(%)	IE(%)	BIES(%)
Ada Boost	89.91	49.85	73.67	60.58	39.79	36.47	77.08
Decision Tree	88.32	78.27	78.53	81.65	73.73	76.67	82.67
KNN	58.45	16.48	22.00	29.81	16.92	22.21	29.81
Random Forest	93.21	87.87	87.03	87.71	87.28	85.89	88.32
Gradiant Boost	92.31	83.46	84.67	85.31	84.18	84.24	86.02
Average	84.44	63.18	69.18	69.01	60.38	61.09	72.78

Table 5
The reported precision score for each classifier and annotation scheme

	IO(%)	IOB(%)	IOE(%)	IOBES(%)	BI(%)	IE(%)	BIES(%)
Ada Boost	96.13	51.54	77.50	68.90	28.06	31.25	86.23
Decision Tree	93.06	81.89	82.67	95.73	77.77	80.73	95.18
KNN	73.43	17.61	32.86	96.88	18.03	33.18	96.88
Random Forest	98.03	91.09	93.33	95.84	90.45	92.38	96.03
Gradiant Boost	97.14	88.77	91.04	94.15	89.26	89.91	95.13
Average	91.55	66.18	75.48	90.30	60.71	65.49	93.89

Table 6
The reported recall score for each classifier and annotation scheme

	IO(%)	IOB(%)	IOE(%)	IOBES(%)	BI(%)	IE(%)	BIES(%)
Ada Boost	84.79	49.32	71.00	56.00	74.90	55.68	69.74
Decision Tree	84.47	75.41	75.14	73.53	70.62	73.43	74.53
KNN	48.72	15.60	16.61	17.95	16.06	16.77	17.95
Random Forest	89.11	85.10	81.99	81.20	84.49	80.74	82.06
Gradiant Boost	88.16	79.44	79.49	78.41	80.28	79.56	79.12
Average	79.05	60.97	64.84	61.41	65.27	61.23	64.68

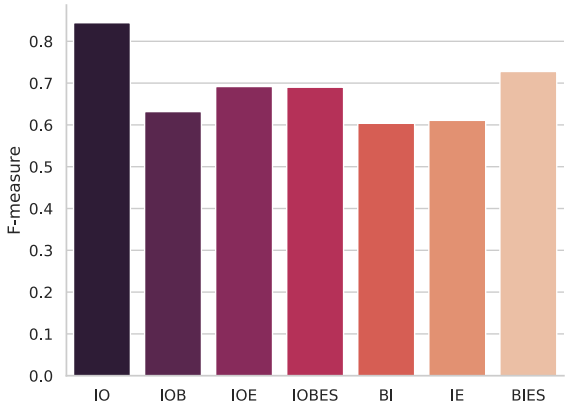

Fig. 3. The reported F-measure scores for each of the studied annotation schemes.

results is necessary, depending on the intended use case or the application. Therefore, the results of these metrics are reported. The IO scheme achieved the highest recall score of 79.05%. On the other hand, the precision score of the IO scheme was not the best, and it came in the second place at 2.34% lower than the best annotation scheme.

From the previous results, we can conclude that the IO scheme achieved the best results among the rest of the reported schemes. Despite that, it is not fair to compare the IO scheme with the others because, it cannot inherently recognize consecutive entities, whereas others do. This issue will be discussed in more detail in Section 4.1.

Without considering the IO scheme, the BIES scheme achieved the highest precision and F-measure score at 72.78% and 93.89% respectively. In contrast, the BIES scheme came in the third place in the recall results with a slightly small difference of 0.59% from the best scheme (BI).

In the literature, the majority of the researchers have paid considerable attention to the type of classifier used in their experiments because the chosen classifier has a significant influence on the performance of the model. While this is a valid consideration, most researchers neglect the type of the annotation scheme, which has been proven to have a significant effect on the results of the classifier, as this study reveals. Regardless of the chosen classifier and to measure the consistency of influence of the chosen annotation scheme on NER, we calculated the arithmetic mean for each classifier using the seven annotation schemes. Then, we compared the results of the classifier for each annotation scheme with the classifier mean. Following this approach will allow us to determine whether a consistent effect exists when choosing a specific scheme despite the chosen classifier.

As it has been seen from the aforementioned Fig. 4, the results generally follow a consistent increasing or decreasing pattern. In the case of using IO, BIES, and IOBES, the results of the models are above the mean regardless of the classifier type. However, the results of IE, BI, and IOB are below the mean. In the case of the IOE scheme, the results are generally below the mean with one exception, the Ada Boost classifier. This led us to conclude that the type of annotation scheme has a consistent influence on the results despite the classifier used.

Furthermore, and in an attempt to examine the performance of the classifier regardless of the chosen annotation scheme, we have calculated the arithmetic mean of each annotation scheme separately. Then, we compared the result of each classifier with the calculated mean. The results of this process is shown in 5. Based on that, it is obvious that choosing the appropriate classifier is a crucial decision to be made, as there are significant differences between the results of the classifiers, regardless of the annotation

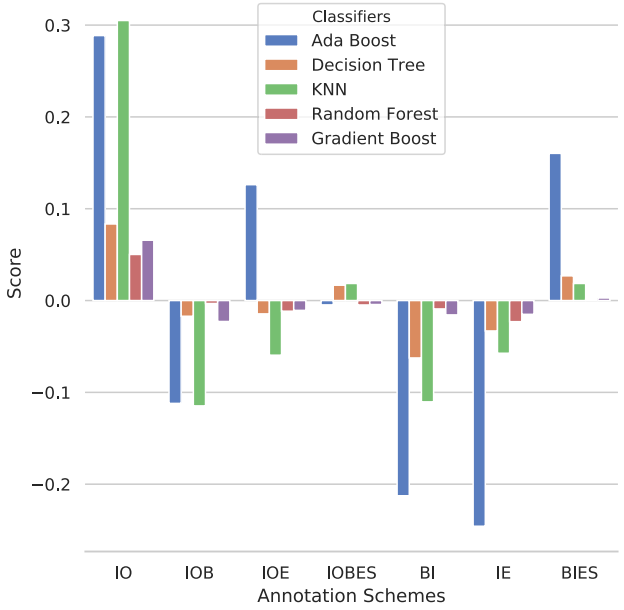

Fig. 4. The difference between the results of the classifiers and the classifiers mean.

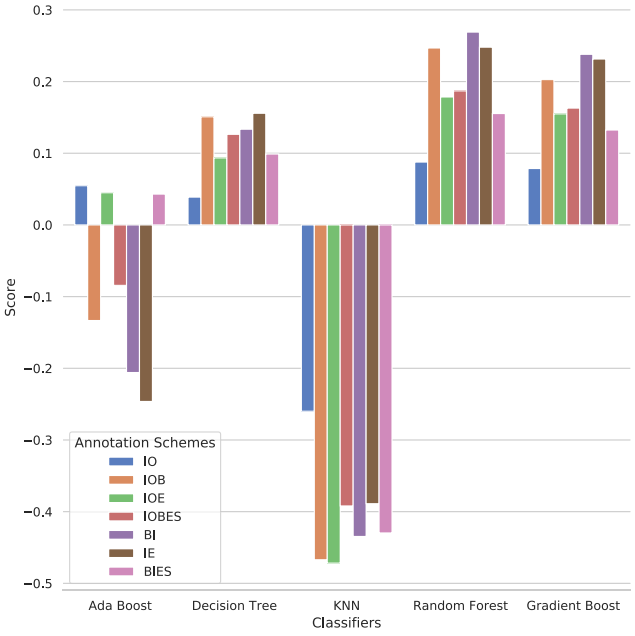

Fig. 5. The difference between the results of the classifiers and the mean of the annotation schemes.

scheme. However, after settling on the appropriate classifier, the next crucial decision to be made, is selecting the suitable annotation scheme because it will negatively or positively affect the performance of the classifier.

4.1. Consecutive entities

As previously shown in Section 4, the IO scheme outperformed the other schemes in terms of recall, precision, and F-measure. However, an important shortcoming of the IO scheme is its inability to recognize consecutive entities. This might have affected the results in favor of the IO scheme, as it lacks the ability to do so because there is only one label assigned to the entities. Thus, con-

Table 7
The percentages of correctly predicted consecutive entities

	IOB(%)	IOE(%)	IOBES(%)	BI(%)	IE(%)	BIES(%)
Ada Boost	31.25	43.75	31.25	31.25	37.50	37.50
Decsion Tree	43.75	43.75	37.50	43.75	43.75	43.75
KNN	12.50	12.50	12.50	12.50	12.50	12.50
Random Forest	50.00	43.75	43.75	62.50	50.00	43.75
Gradient Boost	43.75	62.50	43.75	43.75	50.00	50.00
Average	36.25	41.25	33.75	38.75	38.75	37.50

Table 8
The number of tags needed for each annotation scheme

Annotation Scheme	Number of Tags
IO	C + 1
IOB	2C + 1
IOE	2C + 1
IOBES	4C + 1
BI	2C + 2
IE	2C + 2
BIES	4C + 3

secutive entities blend together. Hypothetically, the IO scheme seems to be the best choice due to several factors: the rarity of the consecutive entities in the corpus, the difficulty of recognizing consecutive entities using other schemes and the fact that it statistically has fewer labels to learn and predict compared to other schemes, which leads to better results.

In our dataset, there are 16 consecutive entities out of 1,278, which constitute less than 2% of the total number of entities. Ensuring sufficient examples in the dataset will lead to better learning and prediction and vice versa. The low number of consecutive entities in our dataset hinders the model. To test the performance of the classifiers in the case of consecutive entities, we carried the experiment illustrated in Table 7, which shows the percentage of correctly recognized consecutive entities per classifier.

Given the results, the performance of the models was promising, although the consecutive entities are rare, as stated earlier. For example, the Gradient Boost classifier in combination with the IOE and the Random Forest classifier with the BI scheme correctly recognized 62.5% of the consecutive entities. The results of other schemes ranged between 12.5% and 50%. This finding shows that these schemes have a promising potential in recognizing consecutive entities, especially if the corpus has many of them.

Given how rare consecutive entities are and how difficult they are to recognize, we assumed that schemes other than IO will not perform well, which makes the IO scheme the preferred choice. However, our findings contradicted this assumption and provided solid results that prove other schemes are capable of predicting these consecutive entities.

4.2. Complexity and running time

One of the shortcomings of the IO scheme is its inability to recognize consecutive entities, as discussed earlier. However, when considering the complexity and running time of the model, using fewer tags to annotate the data decreases the complexity and running time of the model. To elaborate on this point further, the IO scheme outperforms other schemes in terms of cost and running time because it requires fewer tags. Table 8 shows the number of tags needed to annotate the dataset per annotation scheme, where *C* represents the number of categories of named entities. In this research, the value of *C* is constant (C = 1) because we are only recognizing disease names in this experiment. If the task was to recognize more than one category of named entity (e.g., people, location, organizations, etc), the value of *C* would reflect the number of categories of entities which would increase the complexity.

As shown in Table 9, there are significant differences in the average running time for each scheme. As expected, the IO scheme was the fastest. The average execution time was 1.378 s for all classifiers. The average execution time of the rest of the schemes ranged between 2.6 and 3.6 s except BIES, which took an average of 5.5 s to be executed due to its complexity, as stated earlier in Table 8. The reported times are the average running time across the five folds. The execution times were obtained by running on an AMD Ryzen 7 2700X processor with 3.7 GHz operating frequency.

4.3. Comparison with previous studies

In Section 2, we have presented five different previous studies that shed light on the annotation scheme issue. Among these, we chose the study of Cho et al. [8] to compare our results against. The main reason for choosing this study for our comparison can be attributed to the fact that they studied the same annotation schemes as we did. On the other hand, the rest did not cover the majority of the annotation schemes. Even though Konkol and Konopík [2] did present ten annotation schemes, only four are relevant to our study.

Table 10 shows the results of our work against Cho et al. [8] results in terms of the F-measure metric. There are agreements regarding some aspect of the results and some disagreements. Excluding the results of the IO scheme, the best annotation scheme was BIES for both studies. Moreover, the IOBES scheme performed well in both cases. As for the disagreements, the IO annotation scheme achieved the highest score in our study while it achieved the lowest in their study. This difference could be caused by the consecutive entities issue which we discussed in Section 5.

Despite that we compared our results with the results of Cho et al. [8], there are some limitations and shortcomings. The first

Table 9
The execution time for each annotation scheme in seconds

	IO	IOB	IOE	IOBES	BI	IE	BIES
Ada Boost	1.281	1.435	1.404	1.633	1.512	1.531	1.948
Decision Tree	0.137	0.179	0.181	0.187	0.210	0.211	0.232
KNN	0.843	0.935	0.945	0.933	1.048	0.883	0.886
Random Forest	2.174	2.394	2.252	2.226	2.561	2.636	2.918
Gradiant Boost	2.455	8.610	8.650	13.219	11.044	11.426	21.767
Average	1.378	2.7106	2.6864	3.6396	3.275	3.3374	5.5502

Table 10
A comparison of our work with a related study

Scheme	Cho et al.(%)	Our Work(%)
IO	84.63	84.44
IOB	85.81	63.18
IOE	86.05	69.18
IOBES	86.56	69.01
BI	86.25	60.38
IE	85.49	61.09
BIES	86.77	72.78

limitation is that the languages under study are different. We focused on studying Arabic language, while Cho et al. [8] focused solely on the English language. It is worth mentioning that the language itself has an impact on the results as it was evident in the study of Konkol and Konopík [2] where they studied four different languages. The second limitation is that our reported results in Table 10 were based on the average of five different classifiers. On the other hand, Cho et al. [8] experiment relied only on one classifier.

4.4. Limitation and future work

While our research focused on an often neglected area of NER tasks and revealed promising and interesting findings, there are a few concerns and limitations that need further exploration and investigation. Our dataset is devoted to recognizing a single class of entities which is disease names. Despite that we have a sufficient number of disease names, exploring the impact of annotation schemes on other named entities such as drug names and gene names is needed to strengthen the findings and conclusion of this research.

5. Conclusion

In this research, we studied the impact of using different annotation schemes on the performance of NER. Our findings show that the IO annotation scheme, which is the simplest one out of the studied schemes, achieved the highest F-measure score. However, the main limitation of the IO scheme is its inability to recognize consecutive entities. Thus, we explored the ability of more complex schemes to do so. Given the results and the rarity of consecutive entities, the performance of these schemes was promising.

The structure of the language under study constitutes a significant factor that affects the results [2,12,24]. The Arabic language was not heavily studied in the literature for certain aspects, such as annotation schemes, therefore motivating this study. To the best of the authors' knowledge, this study is the first one that has studied the effect of the annotation schemes on the Arabic NER.

References

[1] Sang ETK, Buchholz S, Introduction to the CONLL-2000 Shared Task: Chunking, in: Proceedings of the fourth conference on computational natural language learning and of the second learning language in logic workshop (CONLL/LLL 2000). Lissabon, Portugal, 13–14 september 2000, ACL; 2000. p. 127–32..
[2] Konkol M, Konopík M, Segment representations in named entity recognition. In International conference on text, speech, and dialogue. Springer; 2015. p. 61–70..
[3] Al-Smadi M, Al-Zboon S, Jararweh Y, Juola P. Transfer learning for arabic named entity recognition with deep neural networks. IEEE Access 2020;8:37736–45.
[4] Alkhatib M, Shaalan K. Boosting arabic named entity recognition transliteration with deep learning. In: The thirty-third international flairs conference.
[5] Helwe C, Elbassuoni S. Arabic named entity recognition via deep co-learning. Artif Intell Rev 2019;52(1):197–215.
[6] Obeid O, Zalmout N, Khalifa S, Taji D, Oudah M, Alhafni B, Inoue G, Eryani F, Erdmann A, Habash N. CAMeL tools: an open source python toolkit for arabic natural language processing. In: Proceedings of the 12th language resources and evaluation conference. p. 7022–32.
[7] Mesmia FB, Haddar K, Friburger N, Maurel D. CasANER: Arabic named entity recognition tool. In: Intelligent natural language processing: trends and applications. Springer; 2018. p. 173–98.
[8] Cho H-C, Okazaki N, Miwa M, Tsujii J. Named entity recognition with multiple segment representations. Inf Process Manag 2013;49(4):954–65.
[9] Smith L, Tanabe LK, nee Ando RJ, Kuo C-J, Chung I-F, Hsu C-N et al., Overview of BioCreative II gene mention recognition. Genome Biol 2008;9(2):2008. S2..
[10] Tkachenko A, Petmanson T, Laur S. Named entity recognition in estonian. In Proceedings of the 4th biennial international workshop on Balto–Slavic natural language processing; 2013. p. 78–83..
[11] Malik MK, Sarwar SM. Named entity recognition system for postpositional languages: urdu as a case study. Int J Adv Comput Sci Appl 2016;7(10):141–7.
[12] Mozharova V, Loukachevitch N. Two-stage approach in Russian named entity recognition. In 2016 International FRUCT conference on intelligence, Social Media and Web (ISMW FRUCT), IEEE; 2016. p. 1–6..
[13] Alsughayr A, et al., King Abdullah Bin Abdulaziz Arabic health encyclopedia (www. kaahe. org): A reliable source for health information in Arabic in the internet, Saudi J Med Med Sci 2013;1(1):53..
[14] Alanazi S. A named entity recognition system applied to Arabic text in the medical domain, Ph.D. thesis, Staffordshire University; 2017..
[15] Cohen J. A coefficient of agreement for nominal scales. Edu Psychol Measure 1960;20(1):37–46.
[16] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55(1):119–39.
[17] Breiman L. Classification and regression trees. Routledge; 2017.
[18] Omohundro SM. Five balltree construction algorithms. Int Comput Sci Inst Berkeley 1989.
[19] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[20] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001:1189–232.
[21] Chinchor N, Robinson P. MUC-7 named entity task definition. In Proceedings of the 7th conference on message understanding, vol. 29; 1997. p. 1–21..
[22] Doddington GR, Mitchell A, Przybocki MA, Ramshaw LA, Strassel SM, Weischedel RM. The automatic content extraction (ACE) program-tasks, data, and evaluation. Lrec, Lisbon, vol. 2. p. 1.
[23] Shaalan K. A survey of arabic named entity recognition and classification. Comput. Linguist. 2014;40(2):469–510.
[24] Ahmad MT, Malik MK, Shahzad K, Aslam F, Iqbal A, Nawaz Z, Bukhari F. Named entity recognition and classification for Punjabi Shahmukhi. ACM Trans Asian Low-Res Lang Inf Process (TALLIP) 2020;19(4):1–13.