big data and cognitive computing

MDPI

Article

NERWS: Towards Improving Information Retrieval of Digital Library Management System Using Named Entity Recognition and Word Sense

Ahmed Aliwy [1,*], Ayad Abbas [2] and Ahmed Alkhayyat [3]

[1] Department of Computer Science, University of Kufa, Kufa 54001, Najaf Governorate, Iraq
[2] Department of Computer Science, University of Technology–Iraq, Baghdad 10066, Baghdad Governorate, Iraq; ayad.r.abbas@uotechnology.edu.iq
[3] Department of Computer Technical Engineering, College of Technical Engineering, The Islamic University, Najaf 54001, Najaf Governorate, Iraq; ahmedalkhayyat85@iunajaf.edu.iq
* Correspondence: ahmedh.almajidy@uokufa.com

Abstract: An information retrieval (IR) system is the core of many applications, including digital library management systems (DLMS). The IR-based DLMS depends on either the title with keywords or content as symbolic strings. In contrast, it ignores the meaning of the content or what it indicates. Many researchers tried to improve IR systems either using the named entity recognition (NER) technique or the words' meaning (word sense) and implemented the improvements with a specific language. However, they did not test the IR system using NER and word sense disambiguation together to study the behavior of this system in the presence of these techniques. This paper aims to improve the information retrieval system used by the DLMS by adding the NER and word sense disambiguation (WSD) together for the English and Arabic languages. For NER, a voting technique was used among three completely different classifiers: rules-based, conditional random field (CRF), and bidirectional LSTM-CNN. For WSD, an examples-based method was used to implement it for the first time with the English language. For the IR system, a vector space model (VSM) was used to test the information retrieval system, and it was tested on samples from the library of the University of Kufa for the Arabic and English languages. The overall system results show that the precision, recall, and F-measures were increased from 70.9%, 74.2%, and 72.5% to 89.7%, 91.5%, and 90.6% for the English language and from 66.3%, 69.7%, and 68.0% to 89.3%, 87.1%, and 88.2% for the Arabic language.

Keywords: digital library management system; information retrieval system; named entity recognition; word sense disambiguation

check for updates

Citation: Aliwy, A.; Abbas, A.; Alkhayyat, A. NERWS: Towards Improving Information Retrieval of Digital Library Management System Using Named Entity Recognition and Word Sense. *Big Data Cogn. Comput.* **2021**, *5*, 59. https://doi.org/10.3390/bdcc5040059

Academic Editor: Min Chen

Received: 13 August 2021
Accepted: 19 October 2021
Published: 28 October 2021

1. Introduction

An information retrieval (IR) system is the core of many applications, starting from a simple search engine using exact matching to a complex one using compositional semantics. A digital library management system (DLMS) is a critical application that requires an efficient IR system for retrieving the relevant documents (books, articles, etc.) to the user query. The traditional and old systems of DLMS use an IR system to search for a match based on aspects that include specific keywords, title, author name, year of publication, etc. These systems are very limited and inflexible because they are not indexed according to the content but rather according to a few words. The second category of the DLMS uses an IR system based on all the content of a document. However, these systems still suffer from two types of errors: (i) retrieving many irrelevant documents (false positive error) and (ii) not retrieving many relevant documents (false negative error). These errors result from the nature of natural language. In addition, they demonstrate a certain level of ambiguity at many language levels, such as morphology, syntax, and semantics. In the case of the Arabic language, the level of ambiguity will increase due to the complexity and rich content of Arabic. In recent systems of IR models, several types of semantics are used, such as word sense disambiguation (WSD) and named entity recognition (NER), for performance improvement purposes. In our work, we aim to use these two tasks together, NER and WSD, for improving the IR system used in the DLMS.

A named entity (NE) is a real-world object, such as a personal organization, location, product, etc., represented by a proper name. At the same time, NER is the process of identifying named entities with their types or classes from a predefined set of classes [1,2]. This predefined set of classes can be domain-free (general types), such as a person, organization, etc., or domain-specific, such as a drug or disease in the medical field. NER is the first step in information extraction, and it is useful for many applications, such as question answering (QA), text summarization, trends detection, and so on. Some researchers proved that NER is useful for the IR system and enhances the system's performance for retrieving relevant documents [3].

On the other side, word sense disambiguation (WSD) is the process of selecting the exact meaning of the ambiguous word from predefined sets of senses of this word. Many researchers proved that WSD is useful for the IR system, as will be shown in the Related Works section.

The main problem to be solved in this paper is that the IR systems still suffer from low precision in retrieving what the user wants, and this negatively affects the applications that use these systems, including the DLMS. For example, suppose the user wants to obtain the document that relates to the word Washington (as an organization) in the digital library. In that case, the system will return all the documents that contain this word without distinguishing what the user is looking for (an organization, city, or person); i.e., the system needs a specific NER task. In the case of very rich languages, such as Arabic, there are many synonyms for one meaning of a word, resulting in a weaker IR system if the word sense is not considered.

On the other side, the methods used in Arabic named entity recognition (ANER) and WSD were versions of those used for English, and, in most cases, they do not fit the nature of complex, rich, and highly inflected language in terms of morphology, syntax, and semantics [1].

Furthermore, there are few corpora in the field of ANER, but the freely available ones have many errors at different levels [4] or non-standard corpora [5]. They require manual revision, which, in turn, requires a great amount of time and effort. These reasons cause the researchers in the field of ANER to use a private dataset of a small size.

We propose Arabic and English NER and WSD for the IR model as part of the DLMS, with the aim of (i) improving the work of the DLMS by using NER and semantic facilities for the Arabic and English languages; (ii) increasing the precision, recall, and F-measure by making the DLM system retrieve the relevant documents to the user's query and neglecting the irrelevant ones; (iii) using a suitable approach for the Arabic and English languages since all the digital libraries in the Arab world contain documents of both types.

The contribution of this work can be summarized by:

1. Using a combination of three algorithms of NER suitable for both the Arabic and English languages that are used for the first time.
2. Using NER and WSD with the retrieval system that is used in the DLMS to improve the performance.
3. Providing a manual adjustment of an existing ANER corpus by eliminating many of the bugs in it and adding part of speech (POS) to each word if it does not exist.
4. Testing the efficiency of the system according to the NER, WSD, and IR models.

2. Related Works

There are three levels in our work, which are (i) NER for Arabic and English, (ii) semantic assistance for English and Arabic, and (iii) designing and implementation of the IR system for the DLMS. However, we did not find any previous work that implements all these levels within a whole system. Therefore, the previous works related to these three levels will be discussed in this section.

2.1. NER Stand-Alone Task

For general NER, Zhou and Su [6] proposed a NER system based on the Hidden Markov Model (HMM) with the assistance of privately collected gazetteer lists. They used MUC-6 and MUC-7 datasets for testing the system, and the best results were obtained with the MUC-6 dataset. Chieu and Ng [7] used a maximum entropy-based named entity recognizer (NER) on the MUC-6 and MUC-7 data. The obtained results were close to those obtained by Zhou and Su [6]. Szarvas et al. [8] used AdaBoostM1 and the C4.5 decision tree learning algorithm for the named entity recognition (NER) system with English and Hungarian. Liao and Veeramachaneni [9] used a simple semi-supervised learning algorithm for named entity recognition (NER) with the gold data (annotated manually) of 1000 documents from TF news. Quibaya et al. [10] proposed a combined approach for NER on electronic health records. This approach is a composition of three different methods. They recorded that this approach gave better results than any of the other three methods. Ma and Hovy [11] implemented a combination of bidirectional LSTM, CNN, and CRF as NER on the CoNLL 2003 corpus. Li et al. [12] tried to improve the NER system using a bidirectional recursive network attached with a convolutional network (BRNN-CNN). Devanshu et al. [13] and Sikdar et al. [14] used a conditional random field (CRF) classifier on an English–Spanish dataset. They obtained different results according to the used scenario of testing and preprocessing. Çelebi and Özgür [15] proposed a cluster-based mention type for NER on a private dataset for testing and evaluation. The prediction of a given mention is based on clustered named entities, and then these types are used as features in a ranking model to select the best entity. Yang et al. [16] constructed a NER model with relation extraction based on the BERT language model and Deep Q-Network. They used four public datasets for testing the system. Syed and Chung [17] used Bi-LSTM+CRF with extended feature vectors for NER. They tested the system on a handcrafted food menu corpus from a customers' review dataset.

For Arabic NER, Zaghouani [18] implemented a rules-based ANER system on Arabic news texts. Oodah and Shaalan [19] tried improving a rules-based ANER system and then driving new patterns and testing them using the ACE 2004 NW dataset. El Bazi and Laachfoubi [20] used neural network architecture based on BLSTM and conditional random fields (CRF) for ANER. Liu et al. [21] proposed sequence labeling and ensemble learning for ANER on the AQMAR dataset. Khalifa and Shaalan [22] applied character convolutional neural network (CNN) as augmentation for a NER system and trained on a subset of the Arabic Gigaword corpus. Alkhatib and Shaalan [23] used hybrid deep learning for ANER on ANERCorp and Kalimat corpora. Muhammad et al. [24] used CRF and SVM as an ANER and tested it with the ANERCorp dataset. Al-Smadi et al. proposed transfer learning with deep neural networks for ANER and tested it with the WikiFANEGold dataset. Helwe et al. [25] used a semi-supervised learning approach for an ANER system, and they tested it on three datasets: AQMAR, NEWS, and TWEETS.

All the mentioned works in this section deal with NER as a standalone task regardless of the application. Each researcher's group tested and evaluated the proposed system on a different dataset; therefore, the comparison among their results is difficult.

2.2. Combination of NER and IR System

Few works have been completed for using NER in an IR system. In this section, some of these works will be shown:

Du et al. [26] used NER for improving an IR system. They used two classical NER solutions, which are CRF and topic model-based. Moreover, they processed a very small number of queries for extracting the NEs from them. Krallinger et al. Dalton [27] proposed a system in three levels: (i) named entity recognition, (ii) entity linking, and (iii) ad-hoc document retrieval. Moreover, they extended the dependency-based retrieval models to include structured attributes. Salomonsson [28] used different methods for extracting the NEs and indexing them in a search engine for a complex query. He did not evaluate the proposed IR model. Antony and Mahalakshmi [29] used SVM and DT for extracting and identifying the NE. They used these named entities to improve the biomedical IR system and demonstrated the improvement of the IR system for the Indian language. Krallinger et al. [30] used NER for improving the chemical IR system. Lizarralde et al. [31] presented an approach to enhance web service discoverability that automatically augments web service descriptions. They exploited named entity recognition to identify entities in descriptions and expanded them with information from public text corpora.

All the mentioned works in this section did not deal with the meaning of the word. Moreover, each researcher's group tested and evaluated the proposed system on different datasets.

2.3. Combination Semantic and IR System

Sbattella and Tedesco [32] used a conceptual level and a lexical level as two preprocessing levels for an IR model. The IR model was a stochastic model of a combination of HMM and maximum entropy models that stores the mapping between such levels. Ensan and Bagheri [33] presented a document retrieval model as a semantic-enabled language model based on the semantic relation of the document and query. They tested and evaluated the system on the TREC collections dataset.

El Mahdaouy et al. [34] proposed using semantic similarities of words in existing probabilistic IR models. They tested and evaluated the proposed system on the Arabic TREC collection. Mahmoud and Zrigui [35] proposed a deep learning-based approach for detecting meaning similarity between documents. The relevant features were extracted using the word2vec algorithm and, hence, the sentence vectors representations were produced. They used CNN for estimating the semantic relevancy based on a private dataset.

Jiang [36] proposed semantic information retrieval by combining multiple knowledge sources. They used a labeled dynamic semantic network based on these knowledge sources (Wikipedia, WordNet, and DL ontology) for measuring the semantic relatedness between a query and the documents. Bounhas et al. [37] proposed a morpho-semantic knowledge graph from Arabic vocalized corpora. Both morphological and semantic links were represented through compressed graphs. They evaluated the system in the context of Arabic information retrieval (IR) using several combinations of morpho-semantic query expansion. Mahapatra et al. [38] used a fuzzy cluster-based semantic information retrieval model to determine the exact meaning of the user query. The search engine was used according to the exact meaning for extracting the relevant documents.

These works dealt with the semantic effects on the IR system only regardless of other effects, such as NEs. Moreover, each researcher's group tested and evaluated the proposed system on a different dataset.

Our work is an improvement to the IR model of the DLMS using NER and WSD. It includes improvements of the IR system for the Arabic and English languages.

3. Methodology

In this research, we suggest enhancements for digital library management system (DLMS). These enhancements include adding NER and WSD to the IR system of DLMS. Therefore, our methodology is composed of multiple units: (i) system overview and (ii) system components. Each unit will be explained briefly.

The first part is used for showing the whole system collectively, while the other part is for explaining each step in the system.

3.1. System Overview

Our system has two distinguished subsystems: (i) library initialization system (Figure 1) and (ii) query processing system (Figure 2). These two subsystems have many of the same processes.

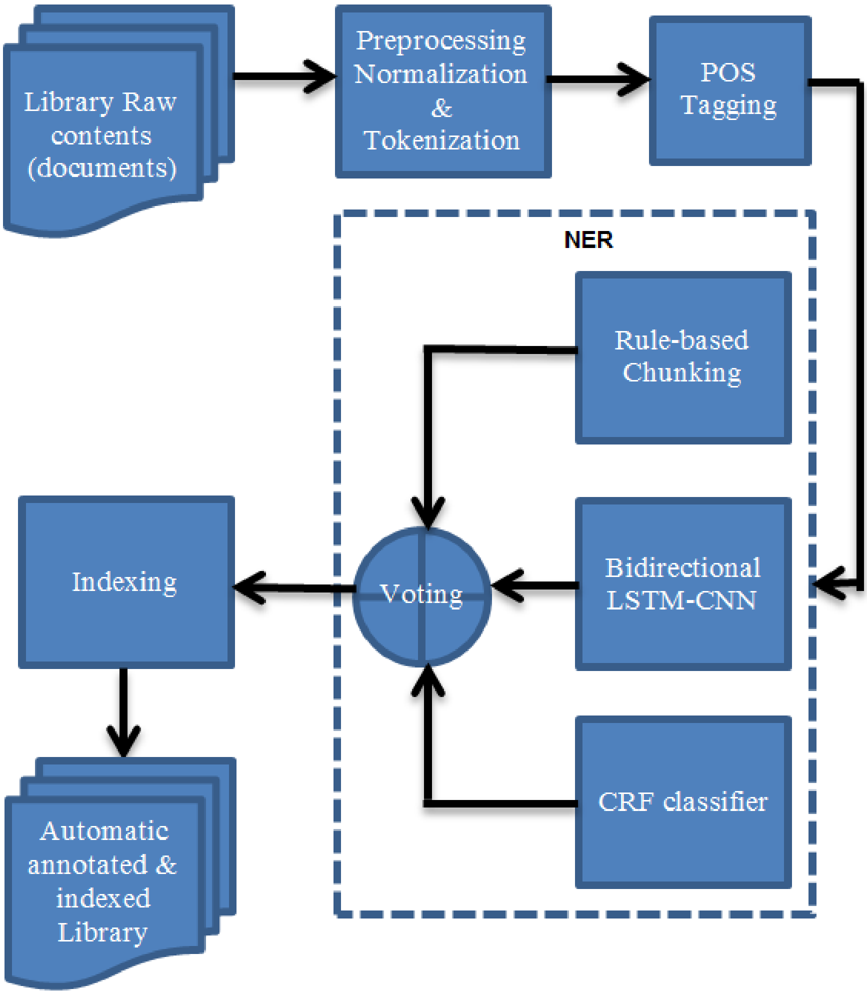

Library Raw contents (documents)
Preprocessing Normalization & Tokenization
POS Tagging
NER
Rule-based Chunking
Bidirectional LSTM-CNN
CRF classifier
Voting
Indexing
Automatic annotated & indexed Library

Figure 1. Library initialization system; the output is automatic annotated & indexed Library.

3.1.1. Library Initialization System

This process is achieved using many sequential processes applied to the contents of the library text. It starts from preprocessing (normalization and tokenization) for extracting the tokens from running text and continues as follows: POS tagging for annotation of the tokens by POS, NER system for extracting NEs with their types, and indexing using the inverted index. The input is raw text, whereas the output is automatically annotated and indexed library content. The process of the library initialization system is shown in Figure 1.

3.1.2. Query Processing System

This process is achieved by many sequential processes on the input query starting from language identification (query is in English or Arabic language) and continues as follows: preprocessing (normalization and tokenization) for extracting the tokens from running text, POS tagging for annotation of the tokens by POS, WSD for extracting the exact meaning for the ambiguous word, query expansion using the synonyms of the exact meaning, NERS for extracting NEs with their types, and IR system for retrieving the most relevant documents for the input query.

The input of this step is a raw text query and the output is the relevant documents to that query. Figure 2 shows this process in detail.

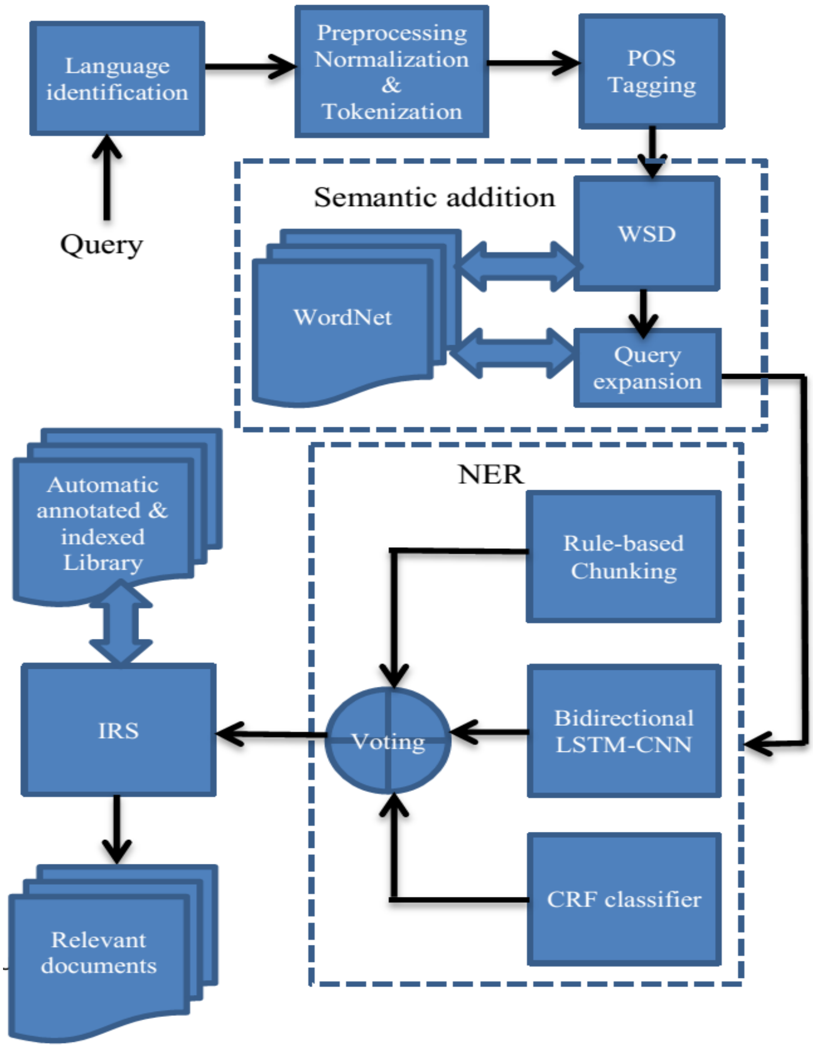


Figure 2. The query processing system; the output is the most relevant document to the query.

3.2. System Components

This section will describe the main components of the proposed system and the purpose and output of each component. In addition, algorithms and methods used within each component of the proposed system will be highlighted in this section.

3.2.1. Language Identification

Most of the library systems in the Arab world deal with documents written in both Arabic and English. Therefore, the language in which the query is written must be known in order to use the proper POS tagging, NER, and WordNet. They are different in the context of English and Arabic language learning.

As two completely different languages are used, the n-gram language model (character level) is sufficient and it is successful even for similar languages [39]. However, any other language can be added to the proposed system according to the requirements, but we should use corpora and WordNet of this language to learn the POS tagger, WSD, and NER. In this work, N-gram language model for language identification is used via the methodology that was used by Selamat [40] with character level.

3.2.2. Preprocessing

The Arabic and English texts should be prepared before feeding them to any applications. This step includes normalization and tokenization process.

Normalization in our context includes (i) tatweel removing (eliminating tatweel symbols from the words), (ii) diacritics removing (eliminating diacritic symbols from the text), and (iii) letter normalization (unification of various forms of letter into unique form). Normalization helps to achieve good accuracy with regard to searching and matching process. A normalization process for Arabic word is shown in Figure 3.

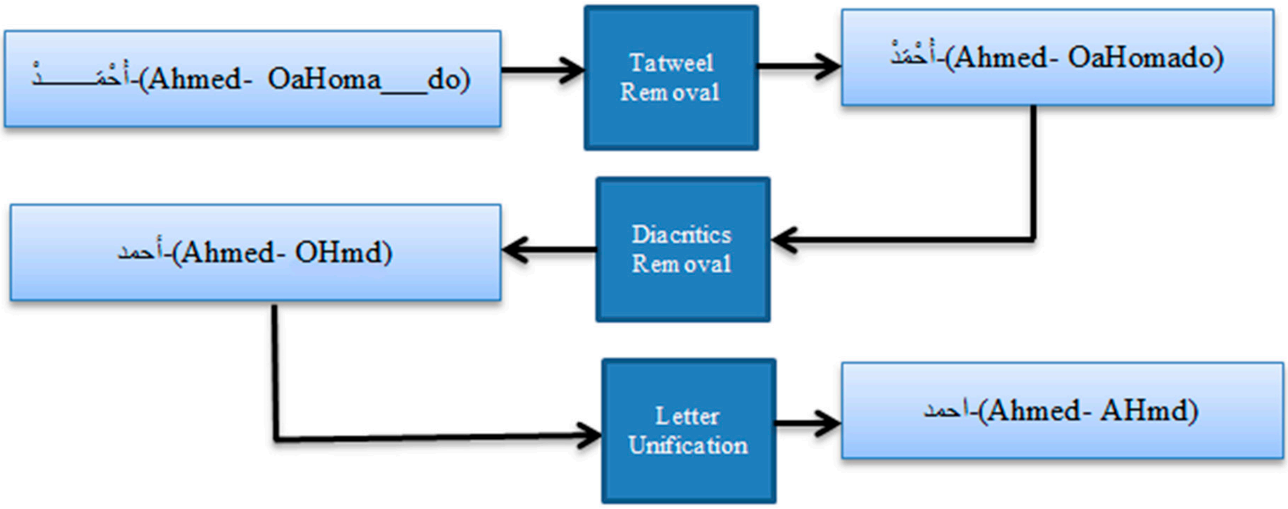


Figure 3. An example of an Arabic word normalization pipeline. Each input/output in the form "Arabic word-(English translation- Buckwalter xml transliteration).

Tokenization is the process of splitting the running text into tokens [1]. The splitting process can be achieved by using white spaces and punctuation marks with few exceptions in the case of numbers, abbreviations, etc. This step is fulfilled regarding English language using Stanford tokenizer as part of Stanford POS tagger [41]. On the other hand, tokenization in the context of Arabic language is achieved using Aliwy tokenizer [42]. The output of this step is the tokens for Arabic and English language.

3.2.3. POS Tagging

POS tagging is the process of assigning part of speech, from a predefined set of POSs, for each token in the sentence according to the context. The output of this step is the sequence of token–POS pairs. For the two languages, the Stanford POS tagger is used. It is a maximum-entropy POS tagger [42] for six languages.

3.2.4. NER

NER is the process of extracting named entities with their types. In this work, a combination of three NERs is used. The three NERs are: (i) rules-based chunk with filtering, (ii) conditional random fields (CRF), and (iii) bidirectional LSTM-CNN. The final NE types are produced by voting among these three types using unweighted voting as shown in Figures 1 and 2.

(i) rules-based chunk with filtering: because the tokens were tagged by POSs in the previous step, the candidate NEs can be extracted using chunk parsing. Then, few different rules, for both Arabic and English languages, have been used to identify the named entity types because they are two different languages in the morphology and syntax levels. The chunk parsing has two types of errors: false positive (extracting candidates that are not named entity) and false negative (does not extract some existing named entity). Therefore, another step is used for filtering the results. This step is used for deciding if the candidate NE is a real NE or not. It can be achieved by using binary classifier learned from the annotated corpus (-1 as not NE and +1 as NE). We used SVM for this purpose with the assumption that the data in the form $x_1y_1 \ldots x_my_m$ where $x_i \in R^n$ is the features vector for the ith sample and $y \in \{-1,+1\}$ represents the class +1 (NE) or -1 (not NE), which can be estimated by:

$$f(x,y,b) = sgn(wx+b) \tag{1}$$

$$w = \sum_{i=1}^{m} y_i, \alpha_i, x_i \tag{2}$$

where: sgn is sign of the value positive or negative, b is the threshold, α_i is weights, and x is the example to be classified.

(ii) CRF: it is a relational learning model and a probabilistic model used to label sequential data. It is used, for NER, to calculate the conditional probability of values as undirected graphical model. If we have a sequence of terms $T = t_1, t_2, \ldots\ t_n$ and their labels $L = l_1, l_2, \ldots\ l_n$, the conditional probability $P(Y \mid X)$ is defined by CRF as follows:

$$\hat{Y} = \underset{y}{argmax} \quad p_\theta(y|x) = \frac{\exp\left(\sum_j w_j F_j(x,y)\right)}{\sum_{\acute{y}} \exp\left(\sum_j w_j F_j(x,\acute{y})\right)} \tag{3}$$

$$F_j(x,y) = \sum_{i=1}^{L} f_j(y_{i-1}, y_i, x, i) \tag{4}$$

where: $\hat{Y}$ is the best label sequence, $p_\theta(y \mid x)$ refers to the probability of calculating a label sequence(y) given a terms sequence(x). $\acute{y}$ refers to all the possible label sequences that can be assigned to a word sequence (sentence). W_j refers to weights assigned to a feature function f_j. The weight vector can be estimated using the limited memory BFGS (L-BFGS) algorithm.

(iii) Bidirectional LSTM-CNN: We followed the methodologies that were used by Collobert et al. [43] for constructing convolutional neural networks (CNN) and that used by Chiu and Nichols [44] for combining bidirectional LSTM with CNN in the stacked method. The CNN is used to extract character-level features, while the sequence-labeling is achieved by BLSTM, which transforms these features to NET scores. This is achieved by estimating two vectors from forward and backward LSTM, and then they are summed for getting the final output value. This methodology was used for Arabic and English language with little difference in the used features.

$$[\hat{y}]_1^T = \underset{[y]_1^T}{argmax} \quad p\left([y]_1^T \middle| [x]_1^T, \widetilde{\theta}\right) = \frac{s\left([x]_1^T \middle| [y]_1^T, \widetilde{\theta}\right)}{\sum_{\forall [j]_1^T} s\left([x]_1^T \middle| [j]_1^T, \widetilde{\theta}\right)} \tag{5}$$

$$s\left([x]_1^T \middle| [i]_1^T, \widetilde{\theta}\right) = \sum_{t=1}^{T} [A]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t} \tag{6}$$

where: $[\hat{y}]_1^T$ represent the best NE tagging sequence for the sentence sequence $[x]_1^T$, $\widetilde{\theta}$ represents set of all parameters to be trained, $A_{i,j}$ represents the score of jumping from tag i to tag j in successive tokens.

(iv) NER features: consideration of specific features for machine learning algorithms is the most important task. In this work, for NER system, we considered some small features, such as: (i) N-grams for letter range from 1 to 3, which is useful for word prefix and suffix, (ii) N-gram for word levels, (iii) if the word contains letters only or alphanumeric, (iv) POS of current word, (v) POSs of the two previous and next words, (vi) if this word part of NE (gazetteers), (vii) if the word starts with a capital letter (English only).

3.2.5. WSD System

Another semantic facility will be added to increase the efficiency of library management system (case of IR model), which is word sense. Extracting the exact meaning of the word is the process of word sense disambiguation (WSD), i.e., WSD is a task of selecting a right sense from a predefined set of word senses according to the context. From an IR point of view, knowing the exact meaning of the words in a query is useful for retrieving the relevant documents to the user query throughout expanding the query according to the synonyms of the word sense. Therefore, WSD is added to our system as another improvement for DLMS. We followed the methodology suggested by Hawraa [45] for English and Arabic WSD, which is used for the first time for English language. The best sense S for the ambiguous word can be estimated by:

$$S = argmax\left(\prod_{t\in sent} D_t \times W_{t,s_i} \times \prod_{t\in win} \text{pos}_{t,s_i}\right) \tag{7}$$

where: D_t represents the weight of the term t in the distance d from the ambiguous word for the input text (query); it is equal to $(1/d + 1)$. POS$_{t,si}$ is the weight of the part of speech POS for term t with sense s_i, and $W_{t,si}$ is the total weight of the term t for sense s_i of the ambiguous word, which can be estimated by:

$$W_{t,s_i} = D_{t,s_i} \sum_{d=1}^{n} W_{t,s_i,d} \times f(t,d,s_i) \tag{8}$$

where $D_{t,si}$ represents the weight of the term t in the distance d from the ambiguous word for sense s in the thesaurus. It is the same as D_t but limited for s_i. Further, $f(t,d,s_i)$ represents the frequency of term t in distance d from the ambiguous word for the sense s_i.

3.2.6. Indexing

Regarding indexing of the terms, the inverted index was used. The inverted index is a structure where each term has a list (posting list) that records which documents the term occurs in. The list in our work has triple tuples: a document ID, frequency of this term in the document, and the positions in the document.

3.2.7. IR Model

We used a simple IR model based on the indexing within the initialization process and the cosine similarity measure between the query and the documents that are represented as vector space model (Equation (9)).

$$sim(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \cdot \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2}\sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \tag{9}$$

$$w_{i,j} = tf_{i,j} \cdot idf_i \tag{10}$$

where d_j is the document j, q is the query, $w_{i,j}$ is the weight of term i in document j, $w_{i,q}$ is the weight of term i in the query q, $tf_{i,j}$ is the term i frequency in document j, and idf_i is the inverse document frequency for the term i.

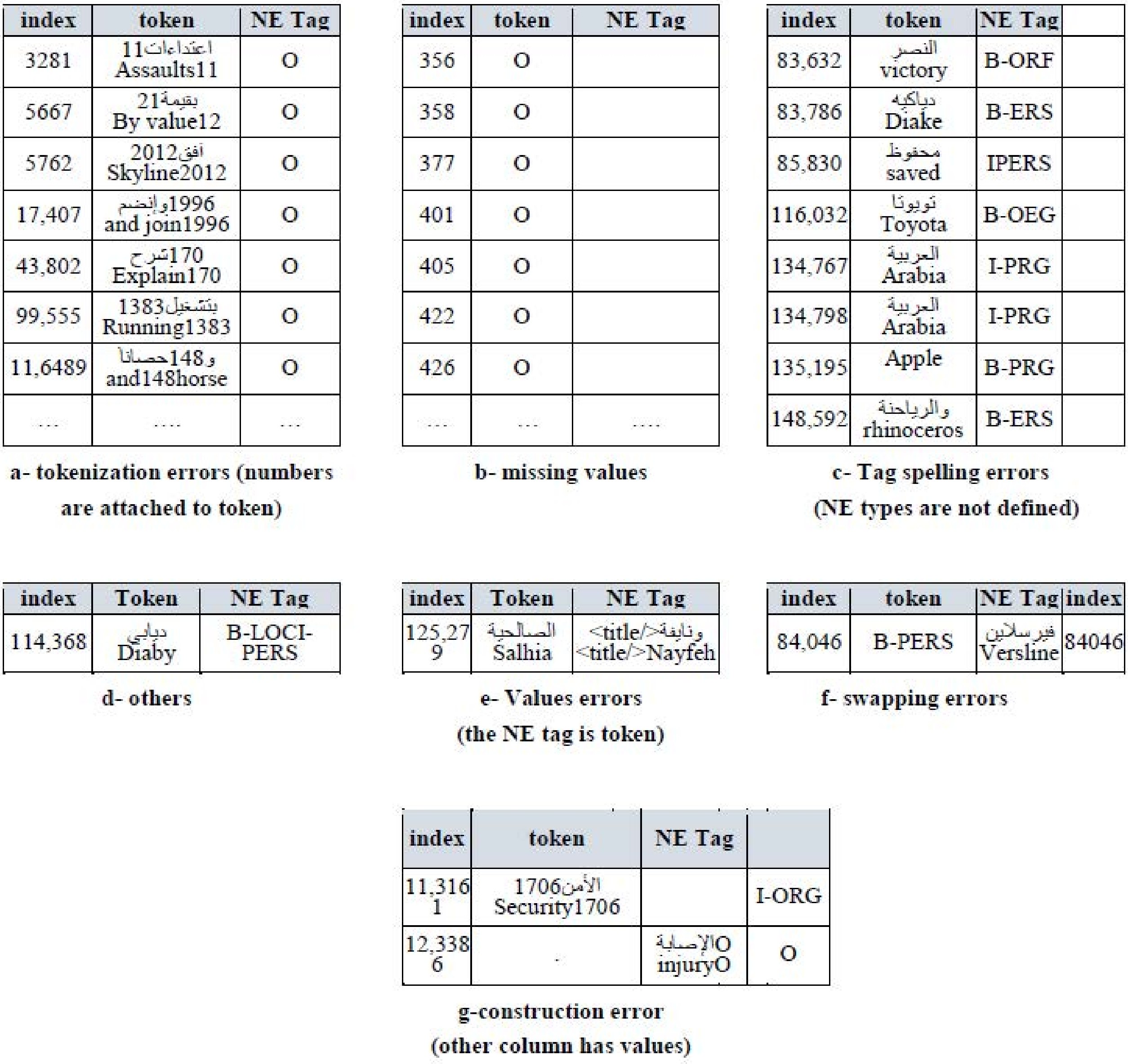
4. Experimental Results and Evaluations

All the experiments were implemented using python 3.7 with some libraries, such as nltk and scikit-learn. The used OS is 64-bit with hardware support of 8 GB memory and an Intel Core i7 processor. This section will show the datasets, results, and evaluation.

4.1. Datasets

Three corpora, annotated with NE, were used in our experiment: the AQMAR and ANERCorp datasets for Arabic; the CoNLL2003 dataset for English. The AQMAR dataset contains text extracted from a small corpus of Arabic Wikipedia articles and hand-annotated for named entities. It has 73,853 annotated tokens [5]. The ANERCorp dataset is an annotated dataset provided by Yassine Benajiba [4]. It has 148,568 tokens after removing the null values. The CoNLL2003 dataset is English data from the CoNLL2003 shared task. It has 302,811 annotated tokens of training, test, and development sets [46]. The Arabic corpora were not annotated with POS tags; therefore, they cannot be used for the evaluation of the POS tagging process. For this reason, we used the corpus used by

Aliwy [47] for the evaluation of POS tagging. Moreover, there are many errors in the Arabic corpora. For example, ANERCorp has errors such as: (i) errors in the structure of the file, (ii) tokenization errors, (iii) spelling errors in the NE types, (iv) missing values, and (v) merging two lines, as shown in Figure 4. All these errors were corrected manually before completing the learning and evaluation to our system.



index	token	NE Tag
3281	اعتداءات11 Assaults11	O
5667	بقيمة21 By value12	O
5762	أفق2012 Skyline2012	O
17,407	وإنضم1996 and join1996	O
43,802	شرح170 Explain170	O
99,555	بتشغيل1383 Running1383	O
11,6489	و148حصانا and148horse	O
…	….	…

a- tokenization errors (numbers are attached to token)

index	token	NE Tag
356	O	
358	O	
377	O	
401	O	
405	O	
422	O	
426	O	
…	…	….

b- missing values

index	token	NE Tag	
83,632	النصر victory	B-ORF	
83,786	دياكيه Diake	B-ERS	
85,830	محفوظ saved	IPERS	
116,032	تويوتا Toyota	B-OEG	
134,767	العربية Arabia	I-PRG	
134,798	العربية Arabia	I-PRG	
135,195	Apple	B-PRG	
148,592	والرياحنة rhinoceros	B-ERS	

c- Tag spelling errors (NE types are not defined)

index	Token	NE Tag
114,368	ديابي Diaby	B-LOCI-PERS

d- others

index	Token	NE Tag
125,279	الصالحية Salhia	<title/>ونايفة <title/>Nayfeh

e- Values errors (the NE tag is token)

index	token	NE Tag	index
84,046	B-PERS	فيرسلاين Versline	84046

f- swapping errors

index	token	NE Tag	
11,3161	الأمن1706 Security1706		I-ORG
12,3386	.	الإصابةO injuryO	O

g-construction error (other column has values)

Figure 4. Some errors in ANERCorp dataset.

4.2. Results

In this section, the results of our system will be presented for Arabic and English. The results will be separated into parts according to the used levels and steps in the system, such as language identification, POS tagging, WSD, NER, and the IR system that was used for retrieving the relevant documents for the input query.

4.2.1. Language Identification

The language model (character level) was used for the language identification between Arabic and English. Despite the two languages being very different in the letter set, we did not use letter matching to distinguish between them because we tried to build a flexible system that can add other similar languages. The results show that our system can identify Arabic and English language with an accuracy of 100%. This result is logical for the previous reason (the languages are very different in letters set).

4.2.2. POS Tagging

As mentioned previously in the POS tagging section, the Stanford POS tagger was used for Arabic and English. However, the Aliwy tokenizer was used for Arabic. Therefore, the POS was tested for knowing its accuracy before and after using the Aliwy tokenizer. The AQMAR and ANERCorp datasets were not annotated with the POS; therefore, another dataset was used for checking the validity of the POS tagger. We used the dataset that was used by Aliwy [47] for evaluating the Arabic POS tagger. It has 658,010 tokens with 546,075 tokens (compatible with ATB schema) including punctuations. For English, the CoNLL2003 dataset was used for evaluation because it was annotated by POS tags. The results are shown in Table 1. After testing the Stanford POS tagger on these data, it was applied to the AQMAR, ANERCorp, and CoNLL2003 datasets.

Table 1. The performance results of POS tagger.

Language	Number of Tokens	Accuracy without Aliwy Tokenizer	Accuracy with Aliwy Tokenizer
Arabic	546,075	92.2	95.7
English	46,665	96.1	

4.2.3. WSD

We followed the methodology suggested by Hawraa [45] for the English and Arabic WSD. We used the Arabic and English WordNet for Arabic and English, respectively, as it is a dictionary-based method. The examples of WordNet were partitioned into 80% for learning and 20% for testing. The accuracies were 73% and 78% for English and Arabic, respectively.

4.2.4. NER

As was explained previously, three methods were used for the NER task: (i) rules-based chunk with filtering, (ii) CRF, and (iii) bidirectional LSTM-CNN. These methods were combined by an unweighted voting for getting the final NE type. The obtained results for precision (P), recall (R), and F-measure (F) are shown in Tables 2 and 3 for Arabic and English, respectively. From these tables, the result of voting is the best, for both languages, where the F-measure is 79.3% and 81.9% for Arabic and 91.5% for English.

Table 2. The performance results of the NERS for Arabic language.

Dataset	Methods	P %	R %	F %
AQMAR	rules-based with filtering	72.4	71.1	71.7
	CRF	75.6	75.2	75.4
	BLSTM-CNN	77.3	79.4	78.3
	Voting	79.8	78.9	79.3
ANERCorp	rules-based with filtering	74.3	73.8	74.0
	CRF	77.7	76.5	77.1
	BLSTM-CNN	78.1	78.9	78.5
	Voting	83.3	80.6	81.9

Table 3. The performance results of the NER system for English language.

Dataset	Methods	P %	R %	F %
CoNLL2003	rules-based with filtering	85.6	83.7	84.6
	CRF	89.9	88.3	89.1
	BLSTM-CNN	90.6	90.4	90.4
	Voting	92.3	90.8	91.5

Figure 5 shows the comparison of the F-measures that were obtained using four classifiers for the NER task on English and Arabic datasets.

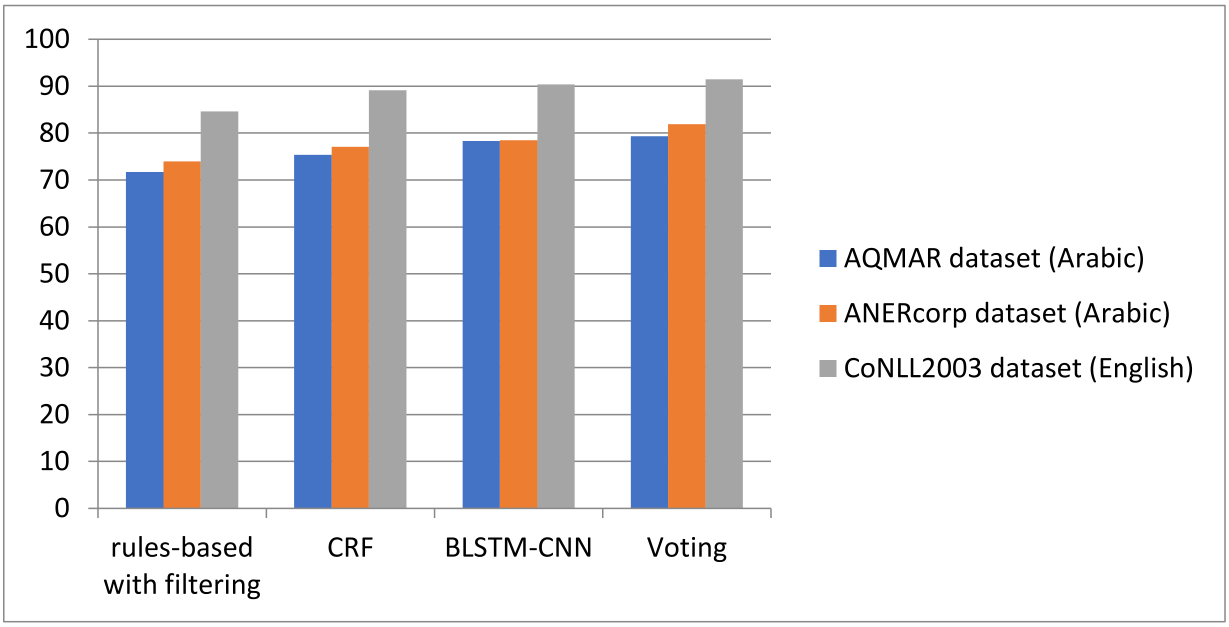


Figure 5. Comparison of F-measures of NER system using four classifiers on English and Arabic datasets.

4.2.5. IR Model

The final testing is the IR system, which represents the overall system test. A small number of documents of the English and Arabic languages were taken. Two hundred Arabic documents and two hundred English documents, from the library of the University of Kufa, were selected for testing with a few different queries in the structure and the meaning. All the queries had at least one NE, and some of them had ambiguous words. The test was done with/without NER and WSD; the results are shown in Tables 4–7.

Table 4. The performance results of IR system without using NER & WSD.

Language	P %	R %	F %
Arabic	66.3	69.7	68.0
English	70.9	74.2	72.5

Table 5. The performance results of IR system with using NER only.

Language	P %	R %	F %
Arabic	86.4	84.7	85.5
English	90.3	88.5	89.4

Table 6. The performance results of IR system with using WSD only.

Language	P %	R %	F %
Arabic	79.8	78.6	79.2
English	72.3	78.2	75.13

Table 7. The performance results of IR system with using NER & WSD.

Language	P %	R %	F %
Arabic	89.3	87.1	88.2
English	89.7	91.5	90.6

Figures 6 and 7 show the comparison of precisions, recalls and F-measures that were obtained in Tables 4–7 for the IR whole system on English and Arabic datasets, respectively.

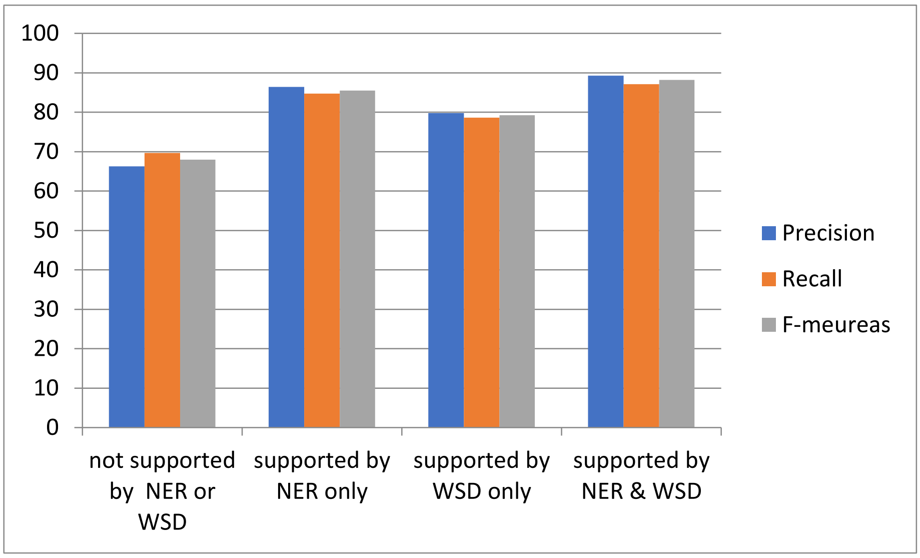


Figure 6. Comparison of precision, recall, and F-measure of IR model that were recorded in Tables 4–7 for Arabic language.

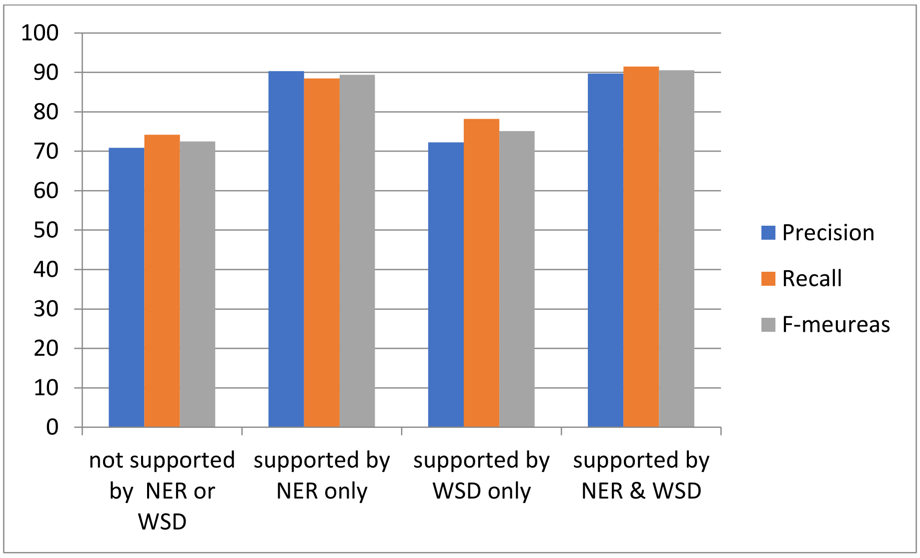


Figure 7. Comparison of precision, recall, and F-measure of IR model that were recorded in Tables 4–7 for English language.

5. Discussion

In this paper, an IR system with its improvements, as part of the DLMS, was implemented. Two distinct processes, for the initialization of the library contents and query processing, were suggested. Several enhancements were added to the IR system as a task of the DLMS. To the best of our knowledge, this is the first work that used NER and WSD collectively for the IR system of Arabic and compared its results with English. All the parts of our system were tested and evaluated for their performance.

For the POS tagging step, the Aliwy tokenizer was used for Arabic to improve the performance of the Stanford POS tagger. The accuracy of the POS tagger was increased by approximately 3.5% from 92.2% to 95.7%. This reveals that the tokenizer included in the Stanford POS tagger has lower performance than the Aliwy tokenizer. Moreover, the accuracy of the Stanford POS tagger for English is better than that for Arabic. This result is logical because the nature of the Arabic language is more complex than the English language in all the natural language processing (NLP) levels [48].

When WSD was used alone as an improvement to the IR, the F-measures were increased for both languages, but, for English, the increase was slightly less than Arabic. We did not find a logical reason for that because the used WordNet for English is more accurate and richer than the one used for Arabic. When we analyzed some of the synonyms in the English WordNet, we saw that some of the synonyms were not very close to each other. This will affect the query expansion by producing words not close to the exact meaning of the query and, hence, decrease the precision.

When NER was used alone as an improvement to the IR, the F-measures were dramatically increased from 68.0% to 85.5% for Arabic and from 72.5% to 89.4% for English. These results prove that the NER is very useful for the IR system used by the DLMS because these systems significantly used the named entities in the search engine.

When NER and WSD were used as improvements to the IR, the F-measures were dramatically increased from 68.0% to 88.2% for Arabic and from 72.5% to 90.6% for English despite using WSD for English affecting the results, but using NER still had the strongest effect for increasing this percentage. This ensures that using NER will improve the IR system and, hence, any application that uses this task, such as the DLMS.

6. Conclusions

An information retrieval system is the core of many applications; one of them is a digital library management system. In this paper, an enhancement to the IR system was made by adding NER and WSD for English and Arabic. The results of the whole retrieval system, for English, of precision, recall, and F-measure, respectively, without improvements, were 70.9, 74.2, and 72.5, and became 89.7, 91.5, and 90.6 after the improvement (using NER and WSD). The results of the whole retrieval system for Arabic, of precision, recall, and F-measure, respectively, without improvements, were 66.3, 69.7, and 68.0, and became 89.3, 87.1, and 88.2 after the improvement (using NER and WSD). This means that the enhancement of the system in the F-measure was 18.1% for English and 20.2% for Arabic. This indicates that the use of NER and the exact meanings of words have a significant impact on the IR.

From the results in Tables 1 and 2, the method used for NER was suitable for both English and Arabic. Moreover, it is clear that voting for NER, among three very different classifiers in the methodology, gave the best results compared to these classifiers. Our NER is more accurate for the English language results as the complexity of this language is less than the complexity of the Arabic language.

For WSD, the English and Arabic WordNets need to be pruned for synonyms groups. It can be done by sorting these synonyms according to the closeness of each synonym to the specific word or by giving each synonym a weight reflecting the closeness of this synonym to the specific word. Still, the accuracy of WSD is low for both languages and needs more effort and scientific research in this field.

As future work, we suggest constructing standard corpora annotated by named entities, word senses, and POS tags. Moreover, we suggest applying this system for the remaining United Nations languages, such as Russian, French, Chinese, and Spanish.

Author Contributions: Conceptualization, A.A. (Ahmed Aliwy); methodology, A.A. (Ahmed Aliwy); software, A.A. (Ahmed Aliwy); validation, A.A. (Ahmed Aliwy), A.A. (Ahmed Alkhayyat) and A.A. (Ayad Abbas); formal analysis, A.A. (Ahmed Aliwy), and A.A. (Ahmed Alkhayyat); investigation, A.A. (Ahmed Aliwy); resources, A.A. (Ahmed Aliwy); data curation, A.A. (Ahmed Aliwy); writing—original draft preparation, A.A. (Ahmed Aliwy); writing—review and editing, A.A. (Ahmed Aliwy), A.A. (Ahmed Alkhayyat) and A.A. (Ayad Abbas); visualization, A.A. (Ahmed Aliwy), A.A. (Ahmed Alkhayyat) and A.A. (Ayad Abbas); supervision, A.A. (Ahmed Aliwy); project administration, A.A. (Ahmed Aliwy); funding acquisition, A.A. (Ahmed Aliwy), A.A. (Ahmed Alkhayyat) and A.A. (Ayad Abbas). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data supporting reported results can be found at: http://www.cs.cmu.edu/~ark/ArabicNER/, https://camel.abudhabi.nyu.edu/anercorp/ and https://www.clips.uantwerpen.be/conll2003/ner/ for AQMAR, NERCorp and conll2003 datasets respectively (accessed on 29 September 2020).

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

References

1. Jurafsky, D.; Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*; Prentice Hall: Hoboken, NJ, USA, 2008.
2. 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*, 37736–37745. [CrossRef]
3. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Lingvisticae Investig. Int. J. Linguist. Lang. Resour.* **2007**, *30*, 3–26. [CrossRef]
4. Benajiba, Y.; Rosso, P.; Benedíruiz, J.M. Anersys: An Arabic named entity recognition system based on maximum entropy. In *International Conference on Intelligent Text Processing and Computational Linguistics*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 143–153.
5. Mohit, B.; Schneider, N.; Bhowmick, R.; Oflazer, K.; Smith, N.A. Recall-oriented learning of named entities in Arabic Wikipedia. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Avignon, France, 23–27 April 2012.
6. Zhou, G.; Su, J. Named entity recognition using an HMM-based chunk tagger. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 473–480.
7. Chieu, H.L.; Ng, H.T. Named entity recognition: A maximum entropy approach using global information. In Proceedings of the COLING 2002: The 19th International Conference on Computational Linguistics, Taipei, Taiwan, 24 August–1 September 2002.
8. Szarvas, G.; Farkas, R.; Kocsor, A. A multilingual named entity recognition system using boosting and C4.5 decision Tree learning algorithms. In *Knowledge Science, Engineering and Management, Proceedings of the First International Conference, KSEM 2006, Guilin, China, 5–8 August 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 267–278.
9. Liao, W.; Veeramachaneni, S. A simple semi-supervised algorithm for named entity recognition. In Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, Boulder, CO, USA, 4 June 2009; pp. 58–65.
10. Quimbaya, A.P.; Múnera, A.S.; Rivera, R.A.G.; Rodríguez, J.C.D.; Velandia, O.M.M.; Peña, A.A.G.; Labbé, C. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Comput. Sci.* **2016**, *100*, 55–61. [CrossRef]
11. Ma, X.; Hovy, E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1.
12. Li, P.-H.; Dong, R.-P.; Wang, Y.-S.; Chou, J.-C.; Ma, W.-Y. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 2664–2669.
13. Jain, D.; Kustikova, M.; Darbari, M.; Gupta, R.; Mayhew, S. Simple features for strong performance on named entity recognition in code-switched twitter data. In Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching, Melbourne, VI, Australia, 19 July 2018; pp. 103–109.

14. Sikdar, U.K.; Barik, B.; Gambäck, B. Named entity recognition on code-switched data using conditional random fields. In Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching, Melbourne, VI, Australia, 19 July 2018; pp. 115–119.
15. Çelebi, A.; Özgür, A. Cluster-based mention typing for named entity disambiguation. *Nat. Lang. Eng.* **2020**, 1–37. [CrossRef]
16. Yang, S.; Yoo, S.; Jeong, O. DeNERT-KG: Named entity and relation extraction model using DQN, knowledge graph, and BERT. *Appl. Sci.* **2020**, *10*, 6429. [CrossRef]
17. Syed, M.; Chung, S.-T. MenuNER: Domain-adapted BERT based NER approach for a domain with limited dataset and its application to food menu domain. *Appl. Sci.* **2021**, *11*, 6007. [CrossRef]
18. Zaghouani, W. RENAR: A rule-based Arabic named entity recognition system. *ACM Trans. Asian Lang. Inf. Process. TALIP* **2012**, *11*, 1–13. [CrossRef]
19. Oudah, M.; Shaalan, K. NERA 2.0: Improving coverage and performance of rule-based named entity recognition for Arabic. *Nat. Lang. Eng.* **2017**, *23*, 441–472. [CrossRef]
20. El Bazi, I.; Laachfoubi, N. Arabic named entity recognition using deep learning approach. *Int. J. Electr. Comput. Eng. IJECE* **2019**, 9. [CrossRef]
21. Liu, L.; Shang, J.; Han, J. Arabic named entity recognition: What works and what's next. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 1–2 August 2019; pp. 60–67.
22. Khalifa, M.; Shaalan, K. Character convolutions for Arabic named entity recognition with long short-term memory networks. *Comput. Speech Lang.* **2019**, *58*, 335–346. [CrossRef]
23. Alkhatib, M.; Shaalan, K. Boosting arabic entity recognition transliteration with deep learning. In Proceedings of the Thirty-Third International Flairs Conference, North Miami Beach, FL, USA, 17–18 May 2020.
24. Muhammad, M.; Rohaim, M.; Hamouda, A.; Abdel-Mageid, S. A comparison between conditional random field and structured support vector machine for Arabic named entity recognition. *J. Comput. Sci.* **2020**, *16*, 117–125. [CrossRef]
25. Helwe, C.; Dib, G.; Shamas, M.; Elbassuoni, S. A semi-supervised BERT approach for Arabic named entity recognition. In Proceedings of the Fifth Arabic Natural Language Processing Workshop, Barcelona, Spain, 12 December 2020; pp. 49–57.
26. Du, J.; Zhang, Z.; Yan, J.; Cui, Y.; Chen, Z. Using search session context for named entity recognition in query. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 765–766.
27. Dalton, J. Entity-Based Enrichment for Information Extraction and Retrieval. Doctoral Dissertation, University of Massachusetts Amherst, Amherst, MA, USA, 2014.
28. Salomonsson, A. Entity-Based Information Retrieval. Master's Thesis, Lund University, Lund, Sweden, 2012.
29. Mahalakshmi, G.S. Content-based information retrieval by named entity recognition and verb semantic role labelling. *J. Univers. Comput. Sci.* **2015**, *21*, 1830.
30. Krallinger, M.; Rabal, O.; Lourenço, A.; Oyarzabal, J.; Valencia, A. Information retrieval and text mining technologies for chemistry. *Chem. Rev.* **2017**, *117*, 7673–7761. [CrossRef]
31. Lizarralde, I.; Mateos, C.; Rodriguez, J.M.; Zunino, A. Exploiting named entity recognition for improving syntactic-based web service discovery. *J. Inf. Sci.* **2019**, *45*, 398–415. [CrossRef]
32. Sbattella, L.; Tedesco, R. A novel semantic information retrieval system based on a three-level domain model. *J. Syst. Softw.* **2013**, *86*, 1426–1452. [CrossRef]
33. Ensan, F.; Bagheri, E. Document retrieval model through semantic linking. In Proceedings of the Tenth ACM International Conference on web Search And Data Mining, Cambridge, UK, 6–10 February 2017; pp. 181–190.
34. El Mahdaouy, A.; El Alaoui, S.O.; Gaussier, E. Improving Arabic information retrieval using word embedding similarities. *Int. J. Speech Technol.* **2018**, *21*, 121–136. [CrossRef]
35. Mahmoud, A.; Zrigui, M. Sentence embedding and convolutional neural network for semantic textual similarity detection in Arabic language. *Arab. J. Sci. Eng.* **2019**, *44*, 9263–9274. [CrossRef]
36. Jiang, Y. Semantically-enhanced information retrieval using multiple knowledge sources. *Clust. Comput.* **2020**, *23*, 2925–2944. [CrossRef]
37. Bounhas, I.; Soudani, N.; Slimani, Y. Building a morpho-semantic knowledge graph for Arabic information retrieval. *Inf. Process. Manag.* **2020**, *57*, 102124. [CrossRef]
38. Mahapatra, D.; Maharana, C.; Panda, S.P.; Mohanty, J.P.; Talib, A.; Mangaraj, A. A fuzzy-cluster based semantic information retrieval system. In Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 11–13 March 2020; pp. 675–678.
39. Garg, A.; Gupta, V.; Jindal, M. A survey of language identification techniques and applications. *J. Emerg. Technol. Web Intell.* **2014**, *6*, 388–400.
40. Selamat, A. Improved N-grams approach for web page language identification. In *Lecture Notes in Computer Science*; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2011; Volume 6910, pp. 1–26.
41. Toutanova, K.; Manning, C.D. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Hong Kong, 7–8 October 2000; pp. 63–70.
42. Aliwy, A.H. Tokenization as preprocessing for Arabic tagging system. *Int. J. Inf. Educ. Technol.* **2012**, *2*, 348–353. [CrossRef]

43. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
44. Chiu, J.P.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
45. Taher, H.A. Arabic Word Sense Disambiguation. Master's Thesis, University of Kufa, Kufa, Iraq, 2019.
46. Tjong Kim Sang, E.F.; de Meulder, F. Introduction to the CoNLL-2003 shared task: Language independent named entity recognition. In Proceedings of the Conference on Natural Language Learning (CoNLL 2003), Edmonton, AB, Canada, 31 May–1 June 2003; Volume 4, pp. 142–147.
47. Aliwy, A.H.; Al-Raza, D.A. Part of speech tagging for Arabic long sentences. *Int. J. Eng. Technol.* **2018**, *7*, 125–128. [CrossRef]
48. Habash, N.Y. Introduction to Arabic natural language processing. *Synth. Lect. Hum. Lang. Technol.* **2010**, *3*, 1–187. [CrossRef]