# Automatic Linking of Short Arabic Texts to Wikipedia Articles

Fatoom Fayad[1*], Iyad AlAgha[2]

[1] Computer Center, Palestine Technical College-Deir El-Balah, Gaza Strip, Palestine.
[2] Faculty of Information Technology, The Islamic University of Gaza, Gaza Strip, Palestine.

* Corresponding author. Tel.: 00972592542727; email: ialagha@iugaza.edu.ps
Manuscript submitted January 10, 2016; accepted March 8, 2016.
doi: 10.17706/jsw.11.12.1207-1223

**Abstract:** Given the enormous amount of unstructured texts available on the Web, there has been an emerging need to increase discoverability of and accessibility to these texts. One of the proposed solutions is to annotate texts with information extracted from background knowledge. Wikipedia, the free encyclopedia, has been recently exploited as a background knowledge to annotate text with complementary information. Given any piece of text, the main challenge is how to determine the most relevant information from Wikipedia with the least effort and time. While Wikipedia-based annotation has mainly targeted the English and Latin versions of Wikipedia, little effort has been devoted to annotate Arabic text using the Arabic version of Wikipedia. In addition, the annotation of short text presents further challenges due to the inability to apply statistical or machine learning techniques that are commonly used with long text. This work proposes an approach for automatic linking of Arabic short texts to articles drawn from Wikipedia. It reports on the several challenges associated with the design and implementation of the linking approach including the processing of the Wikipedia's enormous content, the mapping of texts to Wikipedia articles, the problem of article disambiguation, and the time efficiency. The proposed approach was tested on a dataset of 100 short texts gathered from online Arabic articles. The annotations generated by the approach were compared with the annotations generated by two human subjects. The approach achieved 71.79% accuracy, 74.70% average precision, and 82.63 % average recall. A thorough analysis and discussion of the evaluation results are also presented to address the limitations, strengths as well as recommendations for future improvements.

**Keywords:** Arabic, annotation, entity linking, short text, wikipedia.

## 1. Introduction

This research considers the problem of interlinking unstructured Arabic text to knowledge resources such as Arabic Wikipedia. Linking, or sometimes referred to as entity linking, is the process of linking terms in documents to their corresponding resources from external knowledge bases [1]. For example, terms in Web documents can be converted to hyperlinks leading to Wikipedia articles, so that readers can access complementary information related to the text being read. Linking terms to external knowledge resources provides several benefits such as increased discoverability and accessibility, and hence the utility of information. Furthermore, it can be useful to add semantics to documents and hence allow machines to process documents in an intelligent manner [2].

Manual linking of documents on the Web can be tedious and time-consuming: For each document,

authors or developers should find out relevant resources, e.g. other articles on the Web that contain proper definitions or explanations of the key terms. These difficulties have motivated several researchers to explore approaches for dynamic hyperlinking of text. These approaches have primarily focused on mining the Web for appropriate information to link with terms in the text [3], [4].

With the advent of massive Web-based knowledge bases and encyclopedias such as Wikipedia, DBpedia and Linked Open Data, it has been possible to obtain comprehensive details on almost every topic. These knowledge bases have been used by several efforts to facilitate automatic linking of text [5-8]. Given any piece of text the main challenge is how to determine the most relevant information from knowledge bases with the least effort and time. Existing research has explored the use of different knowledge bases such as DBpedia [8], Wikipedia [9], WordNet [10] and ontologies [2]. Several techniques have been also used to map words in documents to their matching resources/articles from knowledge bases. However, the majority of these efforts have focused on English or Latin-based languages. Little efforts; however, have been done to explore the annotation of Arabic text to external knowledge resource. The limited efforts in this field can be attributed to: 1) The limited number of knowledge bases that are publically-available in Arabic. Although the Arabic versions of Wikipedia and DBpedia are rapidly growing, they are still small when compared to versions in other languages such as English or French. In addition, the number of domain ontologies that are expressed in Arabic is very limited as compared to the ontologies expressed in English. 2) The lack of accurate and efficient Natural Language Processing (NLP) tools for Arabic, as compared to the tools available for English, has notably slowed down research in Arabic NLP in general [11].To our knowledge, Arabic Wikipedia is the largest Arabic knowledge source that covers most of the technical and non-technical topics. Despite its small size, it is the best choice for annotating Arabic text due to its high coverage of many domains. It also supports easy extraction of lexical information due to its relatively high structured content.

This research aims to build an automatic linking approach for short Arabic texts by exploiting Arabic Wikipedia as background knowledge. Given any short Arabic text, the proposed approach searches Wikipedia for the articles that best describe the key terms within the text. It also tries to handle the various challenges associated with the linking process including the processing of the Wikipedia's massive content, the mapping to Wikipedia articles, the ambiguity of terms and the time efficiency. It focuses on short texts because they are generally more difficult to process and annotate than long texts. This is because short texts often do not provide adequate information that enables the application of statistical or machine learning techniques which have been extensively used for the annotation of long texts. For example, it is difficult to apply frequency-based techniques such as TF-IDF to identify keywords in short text. Therefore, the proposed approach will employ alternative techniques to extract keywords from the short text and map them to Wikipedia articles. It can be also applied to the long text by; for example, using a sliding window of a predefined length over the long text.

The work proposed in this paper has the following contributions: First, this is the first work, to our knowledge, that explores the annotation of Arabic text with links to explanatory articles from Wikipedia. Only a few efforts from the literature have tried to interface to the Arabic version of Wikipedia for different purposes such as determining relations between topics [12], named entity recognition [13] and ontology generation [14]. Second, this work presents an in-depth evaluation of the proposed linking approach, and discussion of the potential shortcomings and strengths of each step. This can inform Arab researchers with the various design options and recommendations for developing similar approaches. The source code of the proposed approach and the installation instructions are made available online on (https://github.com/FatoomMFayad/Dynamic-Linking), and are free to use for research and academic purposes.

## 2. Related Works

Several works have explored the annotation of English text with terms extracted from Wikipedia. Wikify [15] is one of the first works that approached this problem by using unsupervised keyword extraction techniques. Their work is limited to the analysis of the document's content without considering the relatedness between entities or their popularity. In addition, Wikify is not adequate for annotating short texts due to its dependency on statistical approaches to extract text features. Milne and Witten [16], [17] extended the original idea by proposing a measure of relatedness that relies on the common links among articles. The proposed measure is used to disambiguate Wikipedia entities by selecting entities that are most coherent with the context of the input text. Kulkarni *et. al* [18] proposed two additional scores to improve the previous approach: The first score models the compatibility between input text and the matching Wikipedia entities. The second score aims at disambiguating entities by measuring the coherence among them. However, finding the best mappings that achieve the highest sum of the two scores is time-consuming. TagMe [19], [20] exploited the relatedness measure introduced by [16] to disambiguate entities, but used additional statistics derived from the pre-processing of Wikipedia. The work in this paper builds on the TagMe approach, and aims to adapt it to the Arabic version Wikipedia. We extended the work of TagMe by using a more simplified yet efficient filter that combines both the coherence and the popularity of Wikipedia entities captured in the text. We also present extensive evaluation results that highlight issues not addressed in previous approaches such as issues related to entity disambiguation and sources of errors.

Another group of works has used graph-based approaches for disambiguation of Wikipedia entities. For example, the AIDA system [21] exploited a new form of coherence-weighted graph , called Mention-Entity Graph, in which nodes are candidate entities while edges are weighted to capture coherence among entities. Authors in [22] used what so called Referent Graph which resembles the Mention-Entity Graph, but uses PageRank algorithm for entity disambiguation. Authors in [23] also used a graph-based approach, but used the HITS algorithm to rank entities. Graph-based approaches can be time-consuming if the graph is dense. In addition, an intensive preprocessing of Wikipedia content is required to construct the weighted graph prior to applying the entity linking approach.

As the amount of Linked Data is rapidly growing, several approaches have been proposed to link text to Linked Data entities such as DBpedia URIs [24]-[26]. Unlike Wikipedia, DBpedia is distinguished by its ontology-based structure that makes the identification of entity associations and properties straightforward. However, the coverage of DBpedia is limited as it does not currently cover the full content of Wikipedia. In addition, the DBpedia's support for some common languages, such as Arabic, is still very limited, hence cannot be used for this work.

The above discussion reveals that the problem of entity linking has been widely explored in previous research using a variety of techniques. However, the use of Arabic Wikipedia for entity linking remains largely unexplored. Existing solutions for English text cannot be applied on Arabic due to the unique characteristics of the Arabic language which require different processing. Recently, there has been a growing interest among Arab researchers to exploit Arabic Wikipedia for different purposes in computer science. Some efforts exploited the semi-structured content of Wikipedia to construct and populate ontologies [14], [27], [28]. These efforts used the Wikipedia' info-boxes, link structure, and pattern matching to extract concepts and semantic relations between words. Other works have exploited Wikipedia features and structure to build Arabic named entities corpuses [13], [29], [30] or to disambiguate named entities [31]. Wikipedia-based categories have been also exploited to improve the categorization of Arabic text [32]. Some works have also used Arabic Wikipedia as background knowledge to expand queries submitted to search engines or question answering systems[33]. The work in this paper adds to previous knowledge by extending the use of Arabic Wikipedia to include the entity-linking problem.

## 3. Setting Arabic Wikipedia

Before elaborating on the linking approach, this section briefly introduces the configuration needed for accessing Wikipedia content. The configuration process is depicted in Figure 1, and includes the preprocessing of Arabic Wikipedia to enable for rapid information access and retrieval. Note that the configuration process is carried out only once.

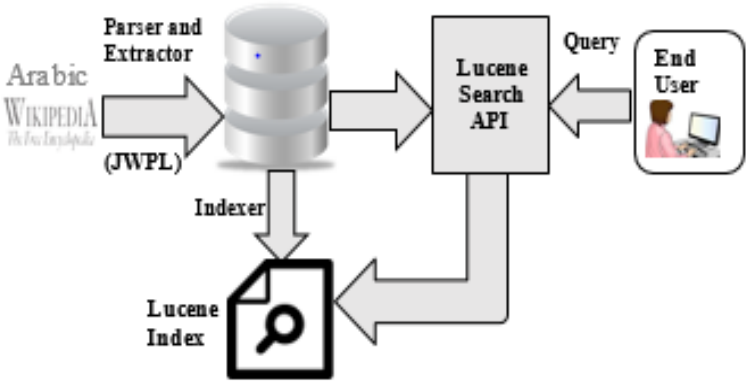


Fig. 1. Setting up Arabic Wikipedia.

To deal with Arabic Wikipedia and extract information from it, we cannot rely on the online version as this will be time-consuming. Therefore, we opted to work offline by downloading the XML dump of Arabic Wikipedia (05 March 2016 version). Information about the downloaded dump is shown in Table 1.

Table 1. Information about the Downloaded Dump

| | |
|---|---|
| XML Dump File Size | 1.57 GB |
| Size after extraction | 22 GB |
| Number of Pages | 869453 |
| Number of Categories | 164497 |
| Number of Categories-inlinks | 506605 |
| Number of Categories-outlinks | 506605 |
| Number of Page-inlinks | 55556636 |
| Number of Page-outlinks | 55556636 |
| Number of Category pages | 4317315 |
| Number of Page-redirects | 9214 |

The XML dump file was then parsed to extract relevant information and store it in a local database. Of the many page attributes available in the dump file, we extracted the page ID, title, content, in-links, and out-links. These are the attributes required to implement the linking approach. Information can be then retrieved by querying the database. JWPL (Java Wikipedia Library) ) [34] was used to parse and extract the previous attributes from the XML dump file. JWPL is an open source API that offers free access to all Wikipedia available information.

The core step of the linking approach is the mapping process, which matches the input Arabic text with Wikipedia entities. The mapping process; however, should be performed rapidly without incurring significant time delays. Therefore, we indexed the Wikipedia page titles, in-links, and out-links by using the Apache Lucene search engine. Apache Lucene [35] is a Java-based search engine library that offers a high-performance cross-platform full-text solution. The content of Wikipedia articles was stored in the local database, but their indices, generated by Lucene, were stored as files on the local machine.

## 4. Linking of Short Arabic Text

The automatic linking approach consists of subsequent steps as depicted in Figure 2. These steps are explained in the following:

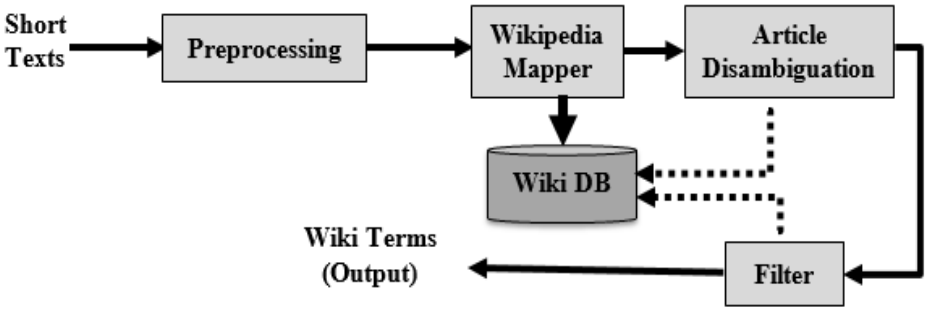


Fig. 2. Dynamic linking process.

### 4.1 Text Preprocessing

We applied the following preprocessing steps on the input Arabic short text: 1) text segmentation, which aims to divide the text into meaningful units. Segmentation of Arabic text is important to split pronouns, conjunctions and prepositions from keywords, thus improving the matching results. 2) Stop-word removal, which aims to remove words that should not be annotated such as prepositions and pronouns. 3) Tokenization and normalization: Normalization aims to unify the Arabic letters that appear in different formats (e.g. replacing "أ"with "ا"and "ه"with "ة"). This enables for better matching with Wikipedia entities. Stanford CoreNLP Toolkit [36] was used for the segmentation and normalization of Arabic text. 4) Generation of n-grams: The input text was then split into a set of n-grams. N-grams are phrases consisting of one or more subsequent words from the input text. The aim of generating n-grams is to increase the matching rate by generating all possible combinations of words and map them to Wikipedia articles. For simplicity, the length of the generated n-grams was limited to three or less, assuming that very few Wikipedia entities will exceed this length. For example, if the input text is " ريال مدريد يفوز على برشلونة في الكلاسيكو", the resulting n-grams will be ريال مدريد يفوز- مدريد يفوز برشلونة- يفوز برشلونة الكلاسيكو – ريال مدريد – مدريد . يفوز – يفوز برشلونة - برشلونة الكلاسيكو- ريال – مدريد – يفوز – برشلونة – الكلاسيكو.

### 4.2 Wikipedia Mapper

The n-grams generated from the preprocessing step are passed to the following component, which is the Wikipedia mapper (see Fig. 2). The mapper is responsible for matching n-grams with titles of Wikipedia articles. The aim is to identify Wikipedia articles that can link to terms in text. N-grams that match with Wikipedia entities will be eventually converted to links to corresponding Wikipedia articles. For each n-gram, the mapper retrieves all Wikipedia articles whose titles contain the n-gram. The mapping process starts with the highest n-grams. The assumption here is that longer phrases will represent more specific descriptors than shorter ones. Then n-grams that are substrings of other longer n-grams are ignored.

### 4.3 Article Disambiguation

When matching phrases to Wikipedia articles, it is likely that a single phrase matches with multiple Wikipedia articles. In fact, the Wikipedia mapper may match a single phrase to a large number of articles. For example, the phrase "ريال مدريد" "Real Madrid"was matched with 43 Wikipedia articles, all of which contain the phrase "ريال مدريد" in their titles. Some of these articles are: بطولات ريال مدريد، قائمة مدربي ريال مدريد،

نادي ريال مدريد، لاعبي ريال مدريد، جماهير نادي ريال مدريد، تاريخ ريال مدريد، قناة ريال مدريد. Furthermore, a term from input text may be ambiguous in the sense that it has multiple meanings. Such an ambiguous term may be mapped to multiple Wikipedia pages, each of which denotes a different meaning. For example, the word "طرابلس" matches with at least two Wikipedia articles, one denotes the Lebanese city while the other denotes the capital of Libya. In case of entity linking, however, a phrase should be linked to a single Wikipedia article. Thus, it is necessary to assure that the detected phrases are linked to most relevant articles among all candidate ones.

When mapping phrases to Wikipedia articles, our assumption is that there should be a collective agreement or relatedness among the detected articles. If a term is mistakenly mapped to an invalid article, this article is likely to have a low relatedness with other detected articles. Therefore, when a term is mapped to multiple articles, this ambiguity can be resolved by determining the article that best relates to other articles associated with other terms in the input text. This idea was inspired by existing efforts that measure semantic relatedness between Wikipedia links [16], [19]. The pairwise relatedness between Wikipedia articles can be measured using the following Equation. [16].

$$\text{relatedness}(p_a, p_b) = 1 - \frac{\log(\max(|A|,|B|) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))} \quad (1)$$

where pa and pb are two Wikipedia articles, A and B are the set of all articles that are linked to pa and pb respectively, and W is the set of all Wikipedia pages. Note that the relatedness score depends on the overlap between their in-linking pages, i.e. $|A \cap B||\boldsymbol{A} \cap \boldsymbol{B}|$.

For each ambiguous article, we calculate its relatedness score with each candidate article detected in the input text by using Equation 1. The overall weight of the ambiguous article is the average of pairwise relatedness scores:

$$\text{Weight}(p) = \frac{\sum_{i=0}^{n} \text{relatedness}(p, Ci)}{|C|} \quad (2)$$

where *p* is the Wikipedia article for which we need to calculate the average relatedness, C is the set of all other candidate articles, Ci is a candidate article, and n is the total number of articles detected from the input text. Of all Wikipedia articles mapped to a single phrase, the article with the highest relatedness weight is chosen. This process is repeated for each detected article in the text until selecting the best articles for all phrases.

To illustrate how this step has led to less ambiguous annotations, assume the text:" الأهلي يضمن الفوز بلقب الدوري الإماراتي بعد خسارة مطارده العين من الفجيرة ". The word "العين" can be associated with a variety of Wikipedia articles including the articles titled as "العين" and "نادي العين الإماراتي". However, the latter article was selected because it had the highest relatedness weight based on the context of the input text.

## 4.4 Terms Filtering

The article disambiguation phase generates a set of articles to link to terms in the input text, one article per term. These phrases; however, have to be filtered to discard terms that may be less informative to the user. To illustrate why the term filtering phase is important, assume that the text is " دونالد ترامب يفوز بالانتخابات التمهيدية بولاية مشيغان". The terms: "دونالد ترامب", "مشيغان", and "يفوز" are all associated with Wikipedia articles. While the first two terms are considered relevant to be annotated, the verb "يفوز" does not convey any

relevant information, thus should be discarded.

To detect such irrelevant terms, we utilized two features that indicate the importance of the associated articles. The first feature is the link probability, which means the probability that the term is used as a link in Wikipedia. The more the term is used as a link in Wikipedia, the more importance it gains. For any term a; the link probability, P (a), is calculated as the following:

$$P(a) = \frac{\text{Number of occurrences of a as an anchor}}{\text{Total number of occurrences of a in Wikipedia}} \quad (3)$$

$$wP(a) = \frac{Numberofoccurancesofaasananchor}{Totalnumberofoccurancesofa inWikipedia}$$

Here P (a) is the link's probability.

The other feature used to detect irrelevant terms is the coherence between the term and other terms detected in the input text. Our assumption is that a term gains more importance if it is related to other terms in the short text. In contrast, it will be considered irrelevant or less important if it is not strongly related to the surrounding terms. The coherence between terms can be determined by measuring the relatedness between their corresponding Wikipedia articles. Therefore, we used the relatedness weights calculated from the article disambiguation phase (refer to Equation 2). Finally, the filtering score of a term is calculated using the following weighted measure:

$$F(a) = \alpha C(a) + \beta P(a) \quad (4)$$

Where $\alpha + \beta = 1.0$

where F (a) is the filtering score that denotes the significance of term a, and its value ranges from 0 to 1. P (a) indicates the link probability, while C(a) indicates the coherence score which is the average relatedness between a and other articles detected in the input text. The factors α and β control the contributions of the two scores in the final filtering score. We performed several experiments to determine the best values for α and β, and found that α = 0.7 and β = 0.3 gave the most acceptable results for the evaluation dataset. However, α and β can vary based on the dataset being used. In the evaluation section, we show how the performance is affected by changing α and β. If the filtering score F (a) is less than a predefined threshold its annotation will be neglected. Based on our experiments in the evaluation section, the threshold was set to 0.3 as this value generated the best results.

Referring to the previous example, the terms "دونالد ترامب", and "مشيغان" are considered relevant since their calculated filtering scores are scores 0.75 and 0.71 respectively. The term "يفوز" is filtered out because its filtering score, which is equal to 0.02, is lower than the threshold value.

## 5. Evaluation

This section presents the experiments we conducted with the following objectives in mind: 1) Assess the reliability of the automatic linking approach: we aimed to explore to what extend the proposed approach can accurately link terms in the input text to relevant Wikipedia articles. We were also interested in exploring any potential errors and the rationales behind these errors. 2) Assess the efficiency of the approach, and identify the steps that potentially consume more time than others.

Similar approaches from the state of the art have been often evaluated by being compared to other

approaches [19, 37, 38]. However, we are not aware of any similar approach that utilizes the Arabic version of Wikipedia for automatic linking to compare with. Therefore, we opted to assess our approach by comparing the output of our approach with the annotations made manually by human subjects over the same dataset.

## 5.1. Dataset

We collected a dataset consisting of 100 short Arabic texts. The texts have different types including tweets, Facebook posts, and Telegram channels' feeds. The short texts are categorized as the following: 70 tweets, 21 Telegram feeds, and 9 Facebook posts. Table 2 shows examples of the dataset. The average number of words per text is 10.63 (SD = 4.57).The complete dataset can be downloaded from https://github.com/FatoomMFayad/Dynamic-Linking.

Table 2. Examples of Short Texts in the Dataset

| Text | Source |
|---|---|
| ميسي في مواجهة خيخون رغم الكدمات | Twitter |
| مليشيا الحوثي وصالح تقصف أحياء عدة في الجبهة الشرقية من تعز | Telegram |
| بيج بن تصمت بعد توقف نادر لأشهر ساعة في العالم 160 عاماً من الرنين | Facebook |

## 5.2. Evaluation Process

Two human subjects were recruited to annotate the dataset by linking terms to Wikipedia articles. The two subjects were University lecturers and had long experience in using Arabic Wikipedia. Human subjects worked individually to inspect the short texts in order to identify terms that can be linked to Wikipedia articles. They were also asked to search Wikipedia for the article that best explains each term.

The results of the annotation process were as the following: One subject annotated a total of 225 terms to Wikipedia articles while the other annotated 215 terms. Both subjects agreed over 213, giving a percentage of agreement that equals to 96.8%. The disagreed annotations included either terms that were linked differently by the two subjects (3 terms), or that were annotated by one subject but not the other (7 terms). Disagreed results were then discussed and reappraised to reach a mutually acceptable opinion. The final consensus result consisted of a total of 213 terms associated with Wikipedia articles. Table 3 shows a sample of the annotations collected from the two subjects: the column to the left denotes the number of the short text in the dataset. The second column contains the detected terms. The third column contains links to Wikipedia articles.

Table 3. Depicts How Results Collected from the Human Subjects

| No | Article Title | URL |
|---|---|---|
| 7 | ريال مدريد | https://ar.wikipedia.org/wiki/%D8%B1%D9%8A%D8%A7%D9%84_%D9%85%D8%AF%D8%B1%D9%8A%D8%AF |
| 7 | نادي خيتافي | https://ar.wikipedia.org/wiki/%D9%86%D8%A7%D8%AF%D9%8A_%D8%AE%D9%8A%D8%AA%D8%A7%D9%81%D9%8A |
| 7 | نادي برشلونة | https://ar.wikipedia.org/wiki/%D9%86%D8%A7%D8%AF%D9%8A_%D8%A8%D8%B1%D8%B4%D9%84%D9%88%D9%86%D8%A9 |
| 7 | أتلتيكو مدريد | https://ar.wikipedia.org/wiki/%D8%A3%D8%AA%D9%84%D8%AA%D9%8A%D9%83%D9%88_%D9%85%D8%AF%D8%B1%D9%8A%D8%AF |

| No | Article Title | URL |
|---|---|---|
| 8 | فرانشيسكو توتي | https://ar.wikipedia.org/wiki/%D9%81%D8%B1%D8%A7%D9%86%D8%B4%D9%8A%D8%B3%D9%83%D9%88_%D8%AA%D9%88%D8%AA%D9%8A |
| 8 | نادي روما | https://ar.wikipedia.org/wiki/%D9%86%D8%A7%D8%AF%D9%8A_%D8%B1%D9%88%D9%85%D8%A7 |

Annotations made by human subjects were then compared with the results obtained from our approach. Therefore, the proposed linking approach was executed over all texts in the dataset. A simple user interface was developed to facilitate the annotation process by inputting a short text and getting annotations as output (see Figure 3). Each output annotation consists of a detected term from the input text and a URL leading to its descriptive article in Wikipedia.

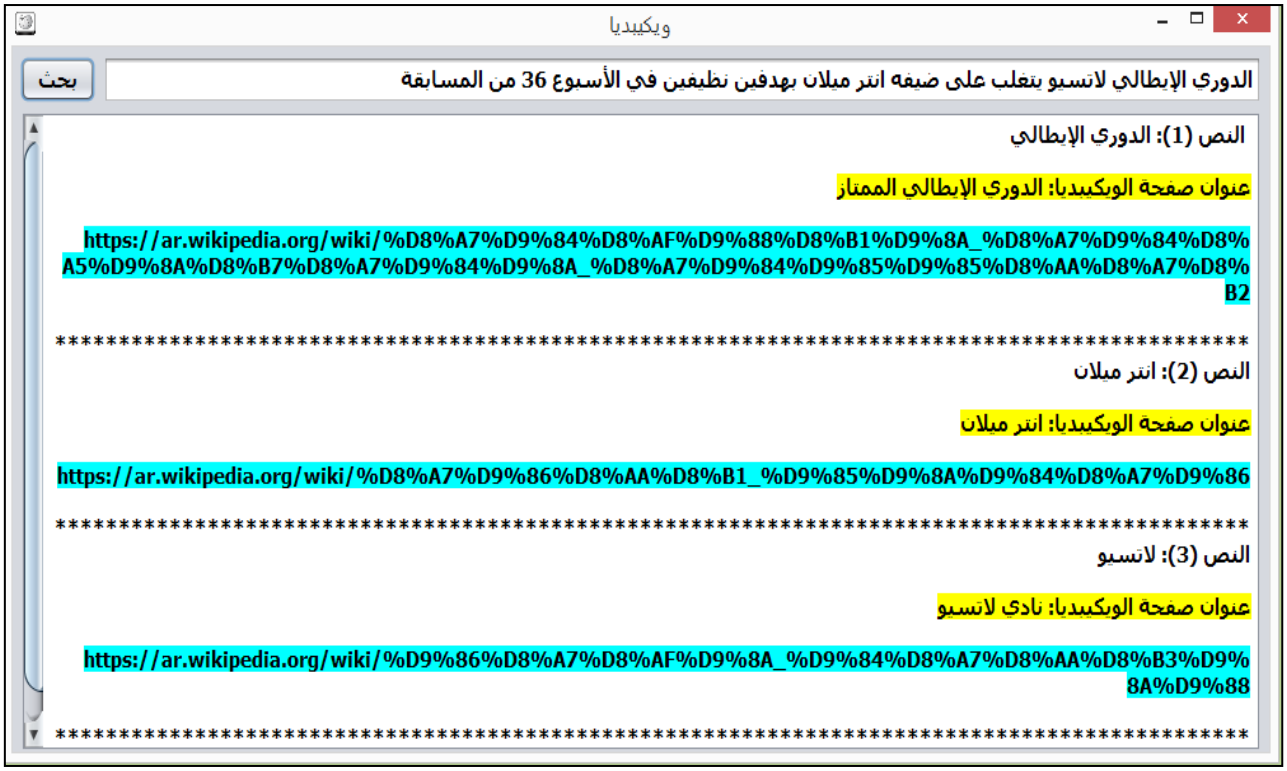

Fig. 3. User interface for entity linking.

## 5.3. Evaluation Metrics

Results were assessed by means of precision, recall and F-measure. Based on previous studies [18], precision and recall were adapted to our experiment as the following:

$$\text{Precision} = \frac{\text{Number of correctly generated links}}{\text{Number of links generated by our approach}}$$

$$\textbf{\textit{Precision}} = \frac{Totalnumberoftermslinkedtorelevant Wikipediaarticles}{Totalnumberoftermslinkedto Wikipediaarticles} \quad (8)$$

$$\text{Recall} = \frac{\text{Number of correctly generated links}}{\text{Number of links made by human subjects}} \quad (9)$$

Note that these metrics were calculated for each short text in the dataset. The overall performance of the approach was estimated by calculating the average precision, recall and F-measure.

## 5.4. Results and Discussion

Table 4 summarizes the evaluation results. The linking approach achieved 74.70% average precision, 82.63 % average recall and 75.21% F-measure. To further explain these results, the generated annotations were inspected thoroughly to identify the main sources of errors and the differences with the annotations made by human subjects. For simplicity, we refer to links made by human subjects as the standard links. Identified errors were classified into the following categories:

**Errors due to Article Disambiguation:** A considerable number of mismatches between the standard links and the generated links are attributed to the variability of the disambiguation process. In some cases, the disambiguation process selected articles that were more specific than the standard articles but inaccurate. For example, in the text "ضغوط أمريكية على آبل لفتح آيفون في الجرائم " the word "أبل" was associated with the article titled as "مطورو أبل" rather than the article titled as "أبل". In other examples, the word "الكرملين" was linked to the article titled as "كأس الكرملين", the word "الفلوجة" was linked to an article titled as: "أحداث الفلوجة", and the word "أمستردام" was mapped to an article titled as: "مؤشر بورصة أمستردام". These specific articles, despite being related to the term in general, do not often give the intended meaning when put in context. They explain sub-topics or events not necessarily related to the input text. This was the most common type of errors, contributing with 60.6 % of the total number of errors generated.

Table 4. Evaluation Metrics of the System

| | |
|---|---|
| Number of Resulted Terms | 235 |
| Macro Average Precision | 74.70 % |
| Macro Average Recall | 82.63 % |
| F-measure | 75.21% |

Surprisingly, the tendency to choose sub- or specific articles sometimes resulted in more accurate links than the standard links. For example, the word "شيعة" in the text: "14 قتيلاً في تفجير استهدف زوار شيعة بالعراق" was associated with the Wikipedia the article titled as "شيعة العراق". This result is more accurate than the human-generated link to the article titled as "الشيعة". In another example, the term "تنظيم الدولة" in the text:" "أميركا ترسل قوات إضافية لمواجهة تنظيم الدولة في العراق" was associated with the article titled as "القاعدة في العراق". While this result is different from the standard link to the article titled as "تنظيم الدولة", it is more related to the context of the text. From the system's point of view, these results can be explained by the fact that specific articles can be more related to the context of text than general articles. This is because specific articles extend the topics of general articles, and thus they are still coherent with the rest of terms in the input text.

A possible solution to the problem of favoring specific articles over general ones is to discard articles with titles that are longer than the target terms. For example, the article titled as "مؤشر بورصة أمستردام" should be discarded as its title is three-word length while the target term, i.e. "أمستردام", is a single word. This solution; however, can cause other errors such as excluding articles with the titles: "جمهورية مصر العربية" and "الولايات المتحدة الأمريكية" for the words: "مصر" and "أمريكا" respectively. A potentially more convenient, though computationally expensive, approach will be to consider the content of articles to resolve the ambiguity.

**Errors due to Term Filtering:** As explained in Section 4.4, term filtering aims to discard terms that are less meaningful to the user. However, two problems with the filtering approach sometimes resulted in the pruning of detected links that should be preserved. The first problem is the low computed filtering scores, thereby causing some important terms to be filtered out. For example, the word "أردوغان" in the text "أردوغان: المنظمات الإرهابية التي تدعي الإسلام أضرت بالإسلام أكثر من الأعداء" was filtered out although it was part of

the standard links. This was due to the low coherence between the word "أردوغان" and other terms in the text. The second problem is that the filtering step depends on a static threshold that can be difficult to determine accurately. As a result, some terms that were considered irrelevant by our human subjects were considered relevant by our approach. These terms had filtering scores slightly over the threshold of 0.3. Consider the following text as an example: " مقال بمجلة نيوزويك الأميركية يتساءل عما فعله الرئيس باراك أوباما تجاه جرائم نظام الرئيس السوري بشار الأسد": the word "مقال" was linked to a Wikipedia article titled as "مقال", and the word "الرئيس" was linked to the article titled as "الرئيس الأمريكي". While these generated links were consistent with the context of the text, they were dismissed by the human subjects as they were seen less important. This type of errors contributed by about 36.36% of all errors.

The term filtering errors; however, do not underestimate the value added by the filtering step which made the results much reasonable for end users. To give a realistic example of the positive impact of the filtering step, consider the following text "قوقل تخطط لعرض الموضوعات الأكثر تداولًا في مربع البحث": Our linking approach detected only a single term in this text, which is "قوقل" and linked it to the relevant article. Without activating the filtering step, the words "قوقل, تخطط, موضوعات, تداول, بحث" were all linked to articles from Wikipedia.

**Errors due to Lack of Semantic Reasoning:** The proposed linking approach primarily relies on the syntactic matching with the titles of articles as well as the link structure of Wikipedia to determine relatedness between articles. However, the lack of semantic inference may result in results that mismatch with the user's interests. For example, in the text "لوبان يتنبأ لابنته بالفشل في انتخابات الرئاسة", the word "لوبان" was linked to the article of Le Pen the daughter rather than the article of Le Pen the father according to the context of the text. This type of errors was the least common among all errors with 3.03%.

## 5.5. Time Efficiency

We evaluated the execution time of the 100 short texts of the dataset. The specifications of the machine used in the evaluation process shown in Table 5.

Table 5. Machine Specification

| | |
|---|---|
| Processor Type | Intel® core i5-5200u |
| Processor Clock Speed | 2.20 Giga Hertz |
| Installed Memory | 8 GB |
| Operating System | Windows 8.1 Enterprise |
| System Type | 64-bit operating system |

Fig. 4 depicts the execution times for the 100 short texts. Table 6 summarizes the results. The average execution time for the 100 short texts was 40.16 seconds. The minimum execution time was 5.927 seconds and the maximum execution time was 235.627 seconds. It is obvious from Fig。 4 that the execution times varied largely (SD = 33.01).

Table 6. Execution Time

| | |
|---|---|
| Average Execution Time | 40.16 seconds |
| Minimum Execution Time | 5.927 seconds |
| Maximum Execution Time | 235.627 seconds |
| Standard Deviation | 33.01 |

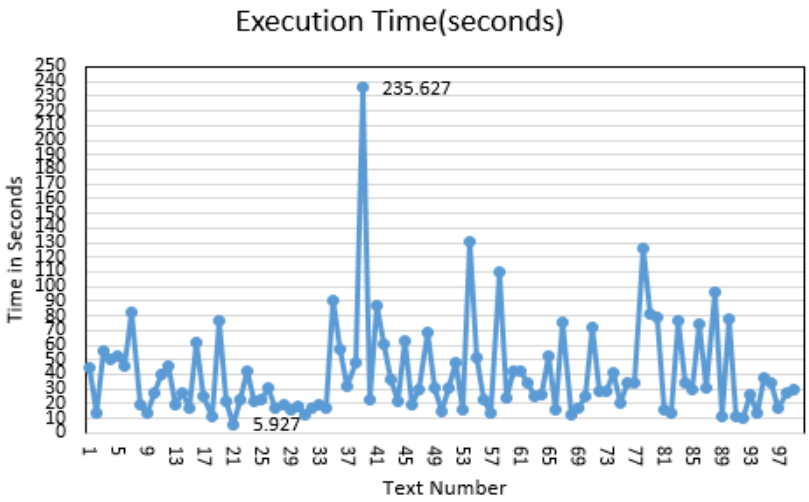


Fig. 4. Execution time for the 100 short texts.

We were interested in identifying what steps of the approach consumed most and least times. Therefore, we measured the time required to execute each of the following steps: the mapping, the article disambiguation, and the filtering steps. Table 7 shows the average execution time of each step.

Table 7. Average Execution Time of the Detailed Steps

| Step | Average Execution Time (seconds) | Standard Deviation |
|---|---|---|
| Mapping | 21.84 | 11.25 |
| Disambiguation | 23.32 | 39.66 |
| Filtering | 1.56 | 1.6 |

These results indicate that the disambiguation step consumed the most time, followed by the mapping step, followed by the filtering step. The time for the mapping step had a low variance across the 100 texts, meaning that it does not significantly vary across texts. However, the long time required for the mapping process was due to the number of search queries executed over the Wikipedia content, which is equal to the number of n-grams generated from the input text. Thus, the mapping process consumes more time as the length of input text increases.

Regarding the disambiguation step, the time required to disambiguate articles varied largely across the short texts. For example, one text took only 1.44 seconds to complete the disambiguation phase, while another text took about 87.08 seconds. This high variance can be explained by the number of ambiguous articles obtained from the mapping step. The larger the number of ambiguous articles, the longer the time needed to handle them. Recall that Equation 1 entails calculating the pairwise relatedness between candidate articles. This means that the complexity of the disambiguation measure increases in a factorial manner as the number of matches with Wikipedia entities increases. Considering that some terms can match with several tens of Wikipedia articles, a large number of calculations should be performed. In contrast, some texts required shorter times due to the small number of ambiguous articles obtained from the mapping step. The analysis of the disambiguation step explains the high fluctuation and variance in the overall execution times of the 100 short texts.

The filtering step consumed the shortest time in comparison with the other steps. Although the filtering step is similar to the disambiguation step in terms of relying on the relatedness measure to estimate the coherence, this step is much faster than the disambiguation step because all ambiguous articles should have been filtered out prior to the filtering step.

Note that our experiments were conducted on a standalone machine, and hence it is possible to improve

the performance further by executing the linking approach over a cluster of computing nodes.

## 5.6 Evaluating the Parameters of Term Filtering

The filtering process explained in Section 4.4 uses a measurement that balances between two features: the link probability and the coherence. In our experiment, the coherence score was weighted more than the link probability (α = 0.7, β= 0.3). The values of the parameters α and β were optimized to achieve the best performance. In addition, the filtering process filters out articles that have filtering score below a predefined threshold. However, these parameters may need to be adjusted based on the nature of the dataset being used. In this section, we explore how the performance of the approach, in terms of F-measure, is affected by adjusting the values of α, β and the threshold. Since the values of α and β should sum to 1, the linking approach was tested on the same dataset while setting α to the values from 0 to 1 with a step of 0.2. Table 8 shows the tested values of α and β along with the resulting performance.

Table 8. Iterations of α

| Iteration | α | β | F-measure |
|---|---|---|---|
| 1 | 0.1 | 0.9 | 0.59 |
| 2 | 0.3 | 0.7 | 0.61 |
| 3 | 0.7 | 0.3 | 0.77 |
| 4 | 0.9 | 0.1 | 0.57 |

Fig. 5 shows how the performance in terms of F-measure changed with α (α = 0.5 was ignored because it returns results that are very close to α = 0.7). The best performance was achieved when α was about 0.7. This value indicates that the best performance was achieved when the coherence was weighted higher than the link probability.

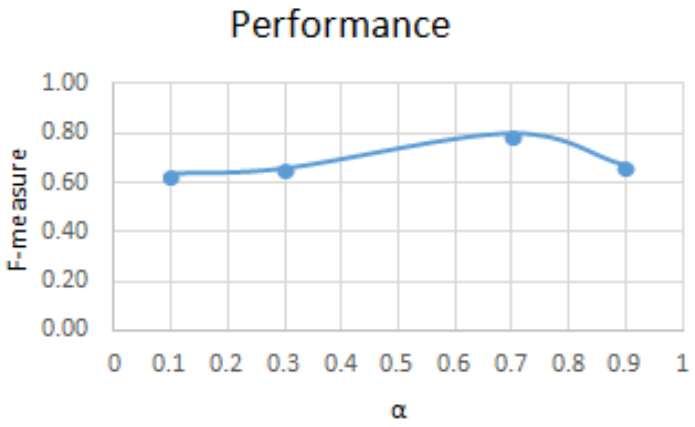


Fig. 5. Effect of α on performance.

Table 9. Effect of Varying Threshold Values on Performance

| Threshold | Precision | Recall | F-measure |
|---|---|---|---|
| 0 | 0.46 | 0.92 | 0.60 |
| 0.2 | 0.67 | 0.97 | 0.76 |
| 0.4 | 0.79 | 0.93 | 0.82 |
| 0.6 | 0.53 | 0.39 | 0.41 |
| 0.8 | 0.37 | 0.15 | 0.21 |

Finally, the effect of varying the threshold on the performance was also assessed. The F-measure score was computed on the same dataset while varying the threshold value from 0 to 0.8 with a step of 0.2. Table 9 and Fig. 6 show the impact of different values of threshold on F-measure. Results showed that the best performance was achieved when the threshold was between 0.2 and 0.4. Then, the performance started degrading as the threshold increased. Therefore, the value of 0.3 was chosen for our threshold, meaning that all articles with filtering scores below this value will be discarded.

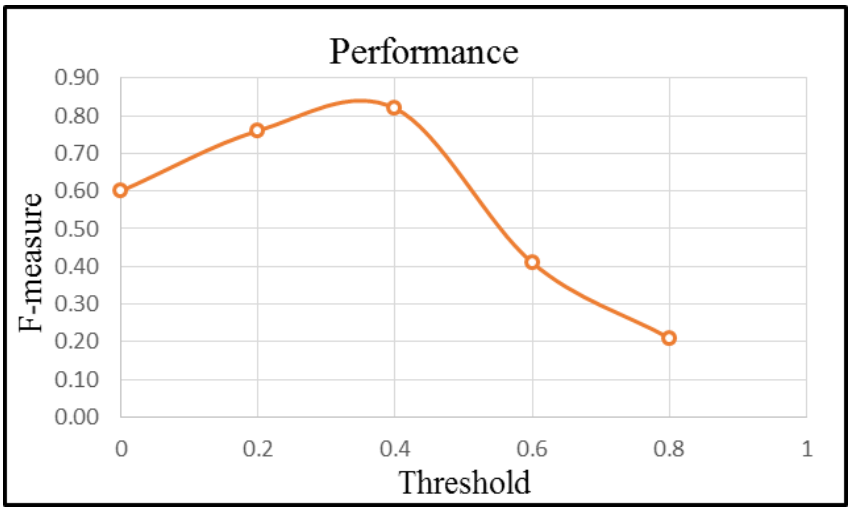


Fig. 6. Effect of threshold on Performance.

## 6. Conclusions and Future Work

In this research, we developed an automatic linking approach for unstructured short Arabic texts by exploiting Arabic Wikipedia as background knowledge. Given an input short text, the approach searches the Wikipedia content for the articles that best describe the most significant terms in the text. The linking approach was evaluated over a dataset of 100 short texts gathered from online resources. Annotations generated by our approach were compared with the annotations made by two human subjects. Results indicated that the approach achieved satisfactory performance that is comparable to the performance of related approaches based on English Wikipedia[19]. The time efficiency was also assessed on a standalone machine whereas the average execution time was 40.16 seconds. Limitations, strengths, and sources of errors were also discussed in detail.

This work is only a first step towards utilizing Arabic Wikipedia for dynamic annotation of text, and there are many dimensions to extend it as the following:

First, alternative ways should be explored to improve the time efficiency: Possible solutions to speed up the linking process may include: 1) Exploring the use of better matching algorithms, and trying to filter generated n-grams before matching them with Wikipedia content. 2) Using multi-processors. 3) Exploring faster approaches to resolve ambiguity among candidate articles.

Second, we will try to improve the article disambiguation process to produce more accurate and intelligent results. Potential solutions may include: 1) Exploiting the content of documents when measuring relatedness between documents (e.g. info boxes, sub-titles) besides the Wikipedia link structure. 2) Exploiting background knowledge such as WordNet and ontologies to boost the documents' relatedness measure. 3) Exploiting natural language processing techniques such as named entity recognition as this will help to identify the potential categories of words to be annotated. 4) Exploring the use of semantic-based techniques to disambiguate terms.

## References


[1] Ramudu, B., & Murty, M. N. (2012). Topic based semantic clustering using Wikipedia knowledge. *Proceedings of the 2012 International Conference on Data Science & Engineering*.

[2] Sriharee, G. (2014). An ontology-based approach to auto-tagging articles. *Vietnam Journal of Computer Science*, *2(2)*, 85-94.

[3] Adnan, M., Warren, J., & M. Orr. (2013). SemLink — Dynamic generation of hyperlinks to enhance patient readability of discharge summaries. *Proceedings of the 26th International Symposium on Computer-Based Medical Systems (CBMS)*.

[4] Navigli, R., & Ponzetto, S. P. (2012). *BabelNet:* The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 217-250.

[5] Geiß, J., Spitz, A., & Gertz, M. (2015). Beyond friendships and followers: The Wikipedia social network. *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*.

[6] Barrena, A., *et al.*, *Combining Mention Context and Hyperlinks from Wikipedia for Named Entity Disambiguation.*

[7] Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*.

[8] Mirizzi, R., *et al.*, Semantic tag cloud generation via DBpedia. *E-Commerce and Web Technologies*.

[9] Ikikat, F. Y., Gurhan, B., & Diri, B. (2015). Automatic linking of wikipedia pages by their semantic similarity. *Proceedings of the International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*.

[10] Passant, A. (2010). Measuring semantic distance on linking data and using it for resources recommendations. *Proceedings of the AAAI Spring Symposium on Linked Data Meets Artificial Intelligence*.

[11] Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*.

[12] Kanan, T., *et al*. (2015). *Extracting Named Entities Using Named Entity Recognizer and Generating Topics Using Latent Dirichlet Allocation Algorithm for Arabic News Articles.* Department of Computer Science, Virginia Polytechnic Institute & State University.

[13] Althobaiti, M., Kruschwitz, U., & Poesio, M. (2014). Automatic creation of Arabic named entity annotated corpus using Wikipedia. *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

[14] Al-Rajebah, N. I., & Al-Khalifa, H. S. (2014). Extracting ontologies from Arabic Wikipedia: A linguistic approach. *Arabian journal for Science and Engineering*, *39(4)*, 2749-2771.

[15] Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. *Proceedings of the sixteenth ACM Conference on Information and Knowledge Management*.

[16] Milne, D., & Witten, I. H. (2008). Learning to link with wikipedia. *Proceedings of the 17th ACM Conference on Information and Knowledge Management*.

[17] Witten, I., & Milne, D. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy.*

[18] Kulkarni, S. *et al*., Collective annotation of Wikipedia entities in web text. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Paris, France: ACM.

[19] Ferragina, P., & Scaiella, U. (2012). Fast and accurate annotation of short texts with Wikipedia pages. *Software*, *29(1)*, 70-75.

[20] Ferragina, P., & Scaiella, U. (2010). Tagme: On-the-fly annotation of short text fragments. *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*.
[21] Hoffart, J., *et al*. Robust disambiguation of named entities in text. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
[22] Han, X., Sun, L., & Zhao, J. Collective entity linking in web text: A graph-based method. *Proceedings of* the *34th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
[23] Usbeck, R. *et al*., *AGDISTIS-Agnostic Disambiguation of Named Entities Using Linked Open Data*.
[24] Mendes, P. N., *et al*. DBpedia spotlight: Shedding light on the web of documents. *Proceedings of the 7th International Conference on Semantic Systems*.
[25] Demartini, G., Difallah, D. E., & Cudré-Mauroux, P. (2012). ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. *Proceedings of the 21st International Conference on World Wide Web*.
[26] Daiber, J., *et al*. Improving efficiency and accuracy in multilingual entity extraction. *Proceedings of the* 9th *International Conference on Semantic Systems*.
[27] Al-Rajebah, N. I., Al-Khalifa, H. S., & Al-Salman, A. S. (2011). Exploiting Arabic Wikipedia for automatic ontology generation: A proposed approach. *Proceedings of the 2011 International Conference on Semantic Technology and Information Retrieval*.
[28] Boudabous, M. M., Belguith, L. H., & Sadat, F. (2013). Exploiting the Arabic Wikipedia for semi-automatic construction of a lexical ontology. *International Journal of Metadata, Semantics and Ontologies.*
[29] Mohit, B., *et al*. Recall-oriented learning of named entities in Arabic Wikipedia. *Proceedings of the 13th* Conference *of the European Chapter of the Association for Computational Linguistics*.
[30] Alotaibi, F., & Lee, M. G. (2012). *Mapping Arabic Wikipedia into the Named Entities Taxonomy*.
[31] Al-Smadi, M., *et al*. A hybrid approach for Arabic named entity disambiguation. *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*.
[32] Yahya, A., & Salhi, A. (2014). Arabic text categorization based on Arabic Wikipedia*. ACM Transactions on Asian Language Information Processing*.
[33] Mahgoub, A. Y., *et al*. (2014). *Semantic Query Expansion for Arabic Information Retrieval.*
[34] Zesch, T., Müller, C., & Gurevych, I. (20080. *Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary*.
[35] Lucene, A. (2016). Apache Lucene. Retrieved from http://archive.apache.org/dist/lucene/java/
[36] Manning, C. D., *et al*. *The Stanford CoreNLP Natural Language Processing Toolkit*.
[37] Shams, E. S., & El-Beltagy, S. R. (2013). News auto-tagging using Wikipedia. *Proceedings of the 2013 9th International Conference on Innovations in Information Technology.*
[38] Ren, Z., *et al*. (2013). Semantic linking and contextualization for social forensic text analysis. *Proceedings of the Intelligence and* Security *Informatics Conference (EISIC).*

**Fatoom M. A. Fayad Libya** was born in 1988. She received her B.S. in computer systems engineering from Palestine Technical College Deir Elbalah, Palestine in 2010. She worked as a lecturer in the computer program at Palestine Technical College Deir Elbalah, teaching the use of applied programming and algorithms in computer science. She has received her MSc in information technology from the Islamic University of Gaza, Palestine in 2016. She is currently working as a database engineer and programmer in the computer center at Palestine Technical College Deir Elbalah, Palestine. Her research interests are data mining, text mining, and machine learning.

**Iyad M. AlAgha** received his MSc and a PhD in computer science from the University of Durham, the UK. He worked as a research associate in the Center of Technology Enhanced Learning at the University of Durham, investigating the use of multi-touch devices for learning and teaching. He is currently working as an assistant professor at the Faculty of Information technology at the Islamic University of Gaza, Palestine. His research interests are semantic web technology, adaptive hypermedia, human-computer interaction and technology enhanced learning.