big data and cognitive computing

MDPI

Article

Topical and Non-Topical Approaches to Measure Similarity between Arabic Questions

Mohammad Daoud

Faculty of Information Technology, American University of Madaba, Amman 11821, Jordan; m.daoud@aum.edu.jo

Abstract: Questions are crucial expressions in any language. Many Natural Language Processing (NLP) or Natural Language Understanding (NLU) applications, such as question-answering computer systems, automatic chatting apps (chatbots), digital virtual assistants, and opinion mining, can benefit from accurately identifying similar questions in an effective manner. We detail methods for identifying similarities between Arabic questions that have been posted online by Internet users and organizations. Our novel approach uses a non-topical rule-based methodology and topical information (textual similarity, lexical similarity, and semantic similarity) to determine if a pair of Arabic questions are similarly paraphrased. Our method counts the lexical and linguistic distances between each question. Additionally, it identifies questions in accordance with their format and scope using expert hypotheses (rules) that have been experimentally shown to be useful and practical. Even if there is a high degree of lexical similarity between a When question (Timex Factoid—inquiring about time) and a Who inquiry (Enamex Factoid—asking about a named entity), they will not be similar. In an experiment using 2200 question pairs, our method attained an accuracy of 0.85, which is remarkable given the simplicity of the solution and the fact that we did not employ any language models or word embedding. In order to cover common Arabic queries presented by Arabic Internet users, we gathered the questions from various online forums and resources. In this study, we describe a unique method for detecting question similarity that does not require intensive processing, a sizable linguistic corpus, or a costly semantic repository. Because there are not many rich Arabic textual resources, this is especially important for informal Arabic text processing on the Internet.

Keywords: computational linguistics; data mining; Arabic question similarity; STS; question paraphrasing; machine learning; NLP

Citation: Daoud, M. Topical and Non-Topical Approaches to Measure Similarity between Arabic Questions. *Big Data Cogn. Comput.* **2022**, *6*, 87. https://doi.org/10.3390/bdcc6030087

Academic Editors: Miltiadis D. Lytras and Andreea Claudia Serban

Received: 19 July 2022
Accepted: 9 August 2022
Published: 22 August 2022

1. Introduction

It is a significant challenge to determine whether two utterances (lexical units, sentences, questions) are similar using Natural Language Processing (NLP) [1]. Similarity detection may lead to the success and the substantially improved results reported from many NLP engines; examples include Text-based Information Retrieval (IR) [2,3], machine translation (MT) [4], text clustering [5], opinion mining, and sentiment analysis [6–8].

The topic of text similarity has been addressed by many researchers in terms of various aspects. Some approaches focus on strings or sub-sequences of characters' similarity between texts, such as longest common sub-sequence (LCS). Alternatively, other approaches, such as cosine similarity and Jaccard similarity, emphasize the importance of the lexical units, where two utterances are similar if they share common words (lexical units) [9]. These methods are considered to be efficient methods for identifying similarity between utterances based on the shared lexical units.

By comparison, it is difficult to find logical similarities between different utterances using semantic similarity, regardless of whether the texts of the different utterances are really similar to one another [10]. For instance, even though texts differ at the word and

character level, the degree of similarity between them may be determined using a corpus or a semantic network [11].

This article focuses on developing automatic methods to determine the similarity between Arabic interrogative statements. Such methods can improve the quality and accuracy of many applications; for example, question-answering computer systems [12], digital virtual assistants [13,14], and automatic chatting apps (chatbots) [15]. The similarity of questions may be considered a sub-problem of the similarity of texts.

However, many academics believe this to be more difficult due to the fact that the linguistic analysis of questions is more complicated and that they have either a brief or non-existent textual context. Furthermore, by definition, questions are prone to being paraphrased (presented in a variety of textual formats) [16,17].

Given that Arabic is regarded as an under-resourced language (in comparison to English) [18,19], the task at hand becomes more challenging. This is particularly the case considering how difficult the process of extracting semantic data from its textual corpus can be [20]. Limited research initiatives have been devoted to tackling the Arabic question similarity problem, resulting in low-to-average outcomes when compared to languages that have extensive textual resources [17].

In the absence or scarcity of a pertinent semantic corpus for the Arabic language, a rule-based approach for labeling questions should be used [21]. In this article, we propose a hybrid system that utilizes experts' hand-crafted rules and supervised learning with various similarity features to find the similarity of Arabic questions and to detect question paraphrasing.

The aim was to investigate two types of similarities: (1) topical and (2) non-topical. Topical similarity is where the questions are asking about the same topic but not necessarily about the same aspect of that topic. For example, Question 1 = "Arabic:–أين غرقت سفينة التايتنك؟English: Where did the Titanic ship sink?"; and Question 2: "Arabic = ـ متى غرقت سفينة التايتنك؟English: When did the Titanic ship sink?". Both questions are both asking about different aspects of the same topic.

Non-topical similarity focuses on the interrogative tools (words) used to form the question, regardless of the topic of the question. For topical similarity, we use lexical and semantic similarity measures. In particular, we use Normalized Google Distance (*NGD*) [22] for semantic similarity, and we use rule-based approaches to address non-topical similarity.

It is common among researchers in this domain to consider only corpus data-driven algorithms to perform clustering and classification tasks on textual data (including questions) [23]. We believe that this is an important aspect of measuring question similarity. However, without the aid of a corpus, basic and straightforward rules may be hypothesized to improve the processing of the questions and to streamline their categorization.

For example, these two Arabic questions are not similar, despite the fact that they have high character subsequence similarity, high word-to-word similarity, and even high topical semantic similarity, merely because Question 1 is asking about the time and Question 2 is asking about a location:

Question 1 $_{Arabic}$ = متى وقعت معركة الكرامة؟

Question 1 $_{English}$ = *When did the Battle of dignity (Al Karamah) occur?"*

Question 2 $_{Arabic}$ = اين وقعت معركة الكرامة؟

Question 2 $_{English}$ = *Where did the Battle of dignity (Al Karamah) occur?"*

Our approach can detect that Q1 and Q2 are topically similar but are different non-topically speaking.

In this article, we present a comprehensive approach to analyzing Arabic questions, and utilize that approach in Arabic question similarity detection with high accuracy given the limited linguistic resources of the Arabic language.

The structure of this article is as follows: The most pertinent previous research is discussed in Section 2. In Section 3, we present our method to measure topical similarity. Section 4 discusses the proposed non-topical similarity measures. Section 5 outlines our data acquisition

and preparation. In Section 6, we present our experimental results, followed by evaluation and assessment remarks in Section 7. Finally, Section 8 lists our conclusions.

2. Text Similarity Approaches

We can view similarity between utterances as character similarity, lexical similarity, and semantic similarity. The focus of this article (question similarity) is a special case of the above similarities.

Character similarity [24] depends on the character arrangement of the text. As a direct consequence of this, two utterances are identical to one another if they include the same strings and characters. Examples of the most frequently used algorithms for character similarity include:

1. Jaro–Winkler [25]: based on the Jaro distance, which measures the edit distance between strings, it is used in computation linguistics and bioinformatics;
2. Needleman–Wunsch [26]: used mostly in bioinformatics;
3. Longest common sub-sequence (LCS) [27]: used mainly in computational linguistics, bioinformatics, and data compression;
4. Damerau–Levenshtein [28]: based on the Levenshtein distance, which is used in bioinformatics, NLP, and fraud detection.

Character similarity algorithms are rarely used alone to deduce similarity between natural texts because these algorithms can be easily misled by word ambiguity and slight morphological changes at the word level, which is a common phenomenon.

Lexical similarity, by comparison, deals with utterances as words (lexical units) attached to each other using a specific grammar [29]. Common methods for measuring lexical similarity between utterances include:

1. Block distance, also known as the taxicab metric or Manhattan distance [30];
2. Cosine similarity [31];
3. Dice's coefficient [32];
4. Euclidean distance (L2);
5. Jaccard similarity [9].

The last two measures are particularly important, because they are efficient and effective for short text similarity (STS) within the same or a related domain. Their effectiveness also increases when there is no lexical ambiguity. However, ambiguous words or texts will affect these approaches.

Semantic similarity offers a tool to address text ambiguity [33]. Semantic similarity correlates texts (words and sentences) based on their logical (meaning) similarity, rather than their character or lexical similarity. Large textual corpora are often used by semantic similarity methods to infer extra information about the words and phrases. For instance, it may conclude that two words are similar based on their similar textual context.

Common methods for measuring semantic similarity between utterances include:

1. Bidirectional Encoder Representations from Transformers (BERT) [34];
2. Word2Vec [35];
3. Explicit Semantic Analysis (ESA) [36]: a vector-based statistical model;
4. Hyperspace Analogue to Language (HAL) [37]: a statistical model based on word co-occurrences;
5. Pointwise Mutual Information—Information Retrieval (PMI-IR) [38]: a statistical model based on a large vocabulary;
6. Second-order co-occurrence pointwise mutual information (SCO-PMI) [39]: a statistical model based on a large vocabulary;
7. Latent Semantic Analysis (LSA) [40]: a vector-based statistical model;
8. Generalized Latent Semantic Analysis (GLSA) [41]: a vector-based statistical model;
9. Normalized Google Distance (*NGD*) [22]: a statistical model based on a large vocabulary from the Google Search engine;

10. Extracting DIStributionally similar words using COoccurrences (DISCO) [42]: a statistical model based on a large vocabulary.

The above algorithms determine similarity considering word and text collocations, and they need a large and well-maintained textual corpus to function reliably and efficiently.

In order to improve the accuracy and coverage of the semantic similarity engine, a semantic network such as Wordnet [43] is often coupled with it.

In reality, a large number of scholars use Wordnet extensively to calculate similarity, which is regarded as a semantic similarity metric that may be used independently. This is beneficial for languages having huge resources, such as English. (There are 155,327 words in the English version of the WordNet, structured into 175,979 synsets.)

The case of question similarity is special because questions usually have a short and limited context. Hence, determining question similarity is considered a challenging task. The challenge increases for the Arabic language, where semantic similarity algorithms cannot be fully utilized because of the absence of rich textual resources. As a result, here we present a hybrid technique that takes advantage of character similarity, lexical similarity, and semantic similarity, but does not need enormous textual resources, to which access is still thought to be a challenge for poorly resourced languages such as Arabic.

3. Topical Similarity

Topical similarity between questions measures the distance between the topics of the questions regardless of the question type or scope. For example, two questions would be considered similar if they both asked about World War II, regardless of the aspects of World War II that are the subjects of the two questions. To determine topical similarity, our approach extracts features from each question as follows:

1. Text features (characters and lexical features);
2. Semantic features.

Accordingly, we measure distances between the features of a pair of questions. The next subsections provide more details.

3.1. Character and Lexical Similarity of Arabic Questions

Here, we process a pair of Arabic questions (AQ1, AQ2) to determine their textual similarity (string and lexical similarity). We use a number of text similarity metrics, which provide a set of features for each pair. In order to create the set of features that belong to the pair, Algorithm 1 processes AQ1 and AQ2 as follows.

The algorithm analyzes a whole array of question couples, C. It starts by sending each question in every couple to an Arabic text normalizer, followed by a special question normalizer (described in Algorithm 2) that tries to eliminate nonstandard question words. This unifies the questions and removes avoidable variations, which will increase the accuracy of the topical similarity. Algorithm 2 uses a dictionary of nonstandard question words mapped to standard words, for example:

Nonstandard question word: "Arabic: في اي مدينة تقع....؟English: in what city do .... located?".

Standard: "Arabic: اينEnglish: where".

Of course, there is a slight difference in the meaning, but this can be tolerated in comparison to the lexical and string distance between the two question words. The given dictionary is arranged in accordance with the length of the nonstandard inquiry words, allowing the algorithm to change words depending on the matches that are the longest.

Algorithm 1: Main algorithm for processing question pairs

```
QuestionAnalyzer (C [ ])
//C is an array of Arabic questions couple,
//each element of C is a couple AQ1, AQ2
//start of Algorithm 1
        For every couple cd (AQ1, AQ2) in C
            normq1 = Normalize (AQ1)
            normq2 = Normalize (AQ2)
            normqq1 = QNorm (normq1)
            normqq2 = QNorm (normq2)
          bowaq1 = BOW (normqq1)
          bowaq2 = BOW (normqq2)
          neraq1 = NER (AQ1)
          neraq2 = NER (AQ2)
          posaq1 = pos (normqq1)
          posaq2 = pos (normqq2)
      F [ d ] [ ] = {
          lcs (normq1, normq2),
          cosine (bowaq1, bowaq2),
          jac (bowaq1, bowaq2),
          euclidian (bowaq1, bowaq2),
          jac (neraq1, neraq2),
          cosine (neraq1, neraq2),
          jac (posaq1, posaq2),
          cosine (posaq1, posaq2),
          StartingSim (bowaq1, bowaq2),
          EndingSim (bowaq1, bowaq2),
          QWordSim (bowaq1, bowaq2)
}
Return F
//end of Algorithm 1
```

Algorithm 2: Question normalization

```
QNorm (AQ)
//start of Algorithm 2
        input dictionary (nonstand, stand) [ ]
//each entry in the dictionary has a standard question "interrogative" form and a //non-standard form
//entries of the dictionary are ordered in an ascending order, starting with the entries with //the longest number of words
        Foreach entry d (nonstand, stand) of dictionary [ ]
            Replace nonstand with stand in AQ
Return AQ
//end of Algorithm 2
```

As shown in Algorithm 1, after the normalization phase (Arabic and text normalization), many similarity measures are used on all the following forms:

(1) *bowaq1* and *bowaq2*: two sets of bags of words corresponding to the normalized AQ1 and AQ2, respectively;
(2) *neraq1* and *neraq2:* two sets of named entities extracted from AQ1 and AQ2, respectively;
(3) *posaq1* and *posaq2:* two forms representing Part of Speech (PoS) tagging of AQ1 and AQ2.

We used the FARASA Arabic tool [44] for the processing pipeline of the Arabic text of each couple.

In summary, Algorithm 1 produces the features below in correspondence to every couple in C:

1. Longest common subsequence for AQ1, AQ2 (after their text and question normalization);
2. Cosine similarity for AQ1, AQ2 after the normalization of their bag of words (BOW);

3. Jaccard similarity for AQ1, AQ2 after the normalization of their bag of words (BOW);
4. Euclidian distance for AQ1, AQ2 after the normalization of their bag of words (BOW);
5. Jaccard similarity for AQ1, AQ2 after the normalization of their Named Entities;
6. Cosine similarity for AQ1, AQ2 after the normalization of their Named Entities;
7. Jaccard similarity for AQ1, AQ2 after the Part of Speech (PoS) analysis of their normalized form;
8. Cosine similarity for AQ1, AQ2 after the Part of Speech (PoS) analysis of their normalized form;
9. Starting similarity measure that was calculated according to Algorithm 3;
10. Ending similarity measure that was calculated according to Algorithm 4;
11. Question word similarity that was calculated according to Algorithm 5.

The following is Algorithm 3, which calculates the starting similarity measure; it receives the normalized bag of words of a question couple and then returns a score of -1, 0, or 1. If the first two words in bowaq1 and bowaq2 are the same, Algorithm 3 returns 1, and if only the first word is similar, it will return 0. Otherwise, it returns -1.

Algorithm 3: Starting similarity algorithm

```
StartingSim (bowaq1, bowaq2)
//start of Algorithm 3
    If bowaq1_1 = = bowaq2_1 && bowaq1_2 = = bowaq2_2
        Return 1
    Elseif bowaq1_1 = = bowaq2_1
        Return 0
    Else
        Return −1
//end of Algorithm 3
```

The following is Algorithm 4, which calculates the ending similarity measure; it receives the normalized bag of words of a question couple and then returns a score of -1, 0, or 1. If the last two words in bowaq1 and bowaq2 are the same, Algorithm 4 returns 1, and if only the last word is similar, it will return 0. Otherwise, it returns -1.

The advantage of this feature is that certain couples may produce high levels of string and lexical similarity; nevertheless, the dissimilarity of the last few words of the questions may completely alter the questions' meaning.

Algorithm 4: Ending similarity algorithm

```
EndingSim (bowaq1, bowaq2)
//start of algorithm 4
    If bowaq1_n = = bowaq2_n && bowaq1_{n−1} = = bowaq2_{n−1}
        Return 1
    Elseif bowaq1_n = = bowaq2_n
        Return 0
    Else
        Return −1
//end of algorithm 4
```

Algorithm 5 receives the normalized bag of words of a question couple and then returns a score of -1, 0, or 1. It determines similarity by relying on the scope of the question. Therefore, if AQ1 and AQ2 have the same type and scope, it returns 1. If their scopes are related, it yields 0, and if they are wholly unlike, it returns -1. A function called findaqw identifies the question word or words that were used in the question. Section 4, "Non topical similarity," further discusses question types and scopes. This feature is a non-topical feature because it is determined purely based on the question type rather than the "topic" of the question.

Algorithm 5: Question type similarity

```
QWordSim (bowaq1, bowaq1)
//start of Algorithm 5
aqw1 = findaqw (bowaq)
aqw2 = findaqw (bowaq2)
        if the scope of aqw1 and aqw2 is the same
                Return 1
        elseif the scopes of aqw1 and aqw2 are related
                Return 0
        else
                Return −1
//end of Algorithms 5
```

3.2. Semantic Similarity (Normalized Google Distance)

We use Normalized Google Distance, often known as *NGD*, to determine semantic similarity. The Normalized Google Distance (*NGD*) is a semantic similarity metric that is computed based on the quantity of results that are provided by the Google search engine in response to a certain query string.

Words with meanings that are different from one another have a tendency to be farther apart on the Normalized Google Distance scale than phrases that are semantically linked to one another.

To be more exact, we can calculate *NGD* of *t* and *r* (where *t* and *r* are both search terms) according to the following formula:

$$NGD(t, r) = \frac{max\{logf(t), logf(r)\} - logf(t, r)}{logG - min\{logf(t), logf(r)\}} \quad (1)$$

where $f(t)$ is the volume of results produced by a Google search for the term t. The same interpretation applies for $f(r)$, and $f(t, r)$ is the number of hits returned when Google is searched for t and r together. G is the total number of pages indexed by Google. $NGD(t, r)$ will be close to 0 if the terms t and r are related. We use *NGD* for Arabic question couples because it is practically convenient, computationally efficient, and does not require a corpus (unlike most other semantic similarity algorithms).

Algorithm 6 shows the steps towards determining *NGD* similarity.

Algorithm 6: Normalized Google Distance similarity

```
NGDSim(AQ1, AQ2)
//Start of Algorithm 6
        nonQT1 = RemoveQW (AQ1)
        nonQT2 = RemoveQW (AQ2)
        ft = callgooglesearch (nonQT1)
        fr = callgooglesearch (nonQT2)
        ftr = callgooglesearch (nonQT1 + nonQT2)
        G = callgooglesearch ("the")
        sim = (max (log ft, log fr)–log ftr)/log G–min (log ft, log fr))
        return sim
//end of Algorithm 6
```

Algorithm 6 receives a couple of Arabic questions and returns their *NGD* similarity. It should be noted that Algorithm 6 removes question words using the RemoveQW function (which is the opposite of findaqw). The number of results that are returned by a search using the term "the" is used in Algorithm 6 to estimate the total number of pages that Google has indexed.

4. Non-Topical Similarity

In this section, we investigate non-topical similarity (interrogative similarity) between Arabic questions. The focus here is on the interrogative tool that was used to form the question rather than the topic of the question. This can be very helpful in determining the overall distance between the two questions.

Table 1 shows the most important scopes of questions asked in Arabic; each scope is labeled corresponding to one of the potential responses to the question. For example, there is no doubt that the response to a Timex Factoid question is either a time or a date. However, for a question about Location Factoids, the response would be a geographical region or a location. Semantically, the two questions (Timex Factoid, Location Factoid) will probably yield two different answers, and consequently, we can deduce a semantic distance even with the presence of high lexical similarity (topical similarity).

Table 1. Common scopes of Arabic questions.

ID	Scope	Answer	Formal Interrogative Form	Paraphrased Words
L	Factoid-Fact	Location	أين Where	Arabic: في أيّ مكان English: In what/which location Arabic: ما موقع English: What is the location Arabic: شارع/قرية/في اي حي English: in what/which neighborhood/town/street
N	Factoid-Fact	Numeric value	كم How many How much	Arabic: ما عدد English: what is the count Arabic: ما قياس English: what is the count Arabic: ما هو طول English: what is the length
T	Factoid-Fact	Time	أيّان ,متى "when"	Arabic: ما تاريخ English: what is the date Arabic: في ايّ وقت English: at what time
NE	Factoid-Fact	Named Entity	لمن Whose	Arabic: الى من English: for whom Arabic: من هو English: Who is Arabic: لاي English: For whom
NED	Definition	Named Entity	ما ,من What	Arabic: ما تعريف English: what is the definition Arabic: من هو English: Who is
M	Method	Method	كيف How	Arabic: ما هي طريقة English: What is the method Arabic: ما هو وصفة English: What is the recipe Arabic: ما الخطوات English: What are the steps
P	Purpose	Purpose	لماذا Why	Arabic: ما هو السبب English: what is the reason Arabic: ما المسبب English: What causes
C	Cause	Cause	ماذا What	Arabic: ما الذي English: What
L	List	List	عدد ,اذكر List	
YN	Yes/No	Yes/No	هل Is/was/are . . .	Arabic: ء English: interrogative Hamzah

We calculate a similarity metric for two Arabic questions by comparing the scope of the interrogative words in each of the questions (question words). When developing the similarity criteria, we make use of both experimental and theoretical approaches.

For instance, it is obvious that a question about a method that begins with "كيفHow" will not be the same as a question about a Timex Factoid that begins with "متىwhen," and on the basis of this, we can construct the following rule:

If AQ1.sid = M and AQ2.sid = T then aqw1 = −1.

Empirical experiments can validate or invalidate this hypothetical rule. Similar rules can be crafted; for instance, if two questions are of the same scope, then the rule would give them a 1 similarity. Through our experiments, we found that some different scopes had unproven similarity or distance; in such occurrences, rules will give them a score of 0.

5. Data Preparation

To test our proposed approach, we compiled 3382 Arabic questions from the Internet. A total of 2932 Arabic questions were extracted from Ejaaba.com (accessed on 1 February 2022), which is a collaborative Arabic community for answering casual questions. In addition, 450 questions were extracted semi-automatically from various Frequently Asked Questions pages, such as those of United Nations organizations, universities, and NGOs.

The 3382 questions were used to randomly generate 2200 Arabic question pairs. Each couple was labeled as T or F (where T indicates a similarity, and F indicates no similarity). In total, 679 couples were given a T label, and 1518 were given an F label.

It was statistically difficult to find a natural occurrence of T couples. Therefore, most of the T-labeled couples were crafted using various paraphrasing approaches by native speakers. We used the same approach for paraphrasing 150 F couples.

Normalization was performed on each of the couples in the dataset, which comprised 2200 couples. Normalization included Arabic text normalization and Arabic question normalization. Then, Algorithm 1 generated the proposed topical and non-topical features.

The scopes of the 3382 different questions are broken down into their respective distributions in Table 2.

Table 2. The breakdown of the scopes of the 3382 unique questions.

Scope	Number of Questions
T	494
L	446
N	389
NE	152
NED	311
M	440
P	271
C	254
L	108
YN	517

The size of our dataset is larger than (or comparable to) similar Arabic and non-Arabic experiments conducted based on labeled data. Table 3 shows the sizes of the datasets of similar experiments.

Table 3. Sizes of datasets of similar experiments.

Name	Language	Task	Size
SemEval-2017 Task 1 [45]	Multilingual, including Arabic	Semantic Textual Similarity	1101 Arabic pairs
SemEval-2016 Task 3, subtask B [46]	English	Question Similarity	317 original, 1999 Q-Q pairs
SemEval-2022 Task 8 [47]	Multilingual, including Arabic	News Similarity	548 Arabic Pairs
SemEval-2019 Task 8 [48]	English	Question Answering	2310 questions
Nagoudi [49]	Arabic–English	Short Text similarity	2400 English-Arabic pairs

6. Experimentation and Results

The 2200 couples were divided into 1450 couples as a training set and the remaining 750 couples as a test set. Although references do not have a perfect data split ratio between training and test sets, we chose a split ratio of 65.91% training to 34.09% testing in our experiment for the following reasons:

1. The ratio of 60–70% for training is common [6,7] and was successfully used in similar experiments with comparable size and dimensions [8].
2. Many researchers reported that 67% training to 33% test reported optimized results when datasets were small [9].
3. Our split satisfies the ratio suggested by [10] to achieve optimality, which is $\sqrt[2]{p} : 1$, where p is the "effective number of parameters." In our case, this is 4. Therefore, our split should be close to 2:1, which is close to the ratio we used.

The resulting dataset was subjected to a variety of classifiers. We note that these classifiers were selected based on the guidelines outlined in [50].

As shown in Table 4, the Random Forest classifier [51] with a nine-fold cross validation produced the best results in terms of accuracy, recall, and F-measure.

Table 4. Comparison between top average precisions reported by various selected classifiers.

Classifier	Top Average Precision
Random Forest	0.84
REPTree [52]	0.82
ADABoost [53]	0.80
J48 [54]	0.83
Naïve Bayes [54]	0.69
SVM [55]	0.75
ANN (4 dense layers, 20 epochs) [55]	0.81

The outcomes generated by the Random Forest classifier are listed in Table 5.

Table 5. Results from the Random Forest algorithm, using the topical and non-topical features we calculated.

	Precision	Recall	F1
T	84%	59%	70%
F	87%	96%	91%
Average	84%	85%	85%

In order to evaluate our proposed methodology and features, we carried out the experiment without making use of our unique features. This means that we did not use the following features:

(1) EndSim;

(2) StartSim;
(3) QWordSim;
(4) NGDSimilarity

As a result, the evaluation depended only on elementary features extracted by measures such as the cosine similarity measure, the Jaccard distance, the Euclidean distance, and the LCS.

The results of the same test are shown in Table 6, but they do not include our topical or non-topical features.

Table 6. The results that were produced by the Random Forest algorithm, without our special similarity features.

	Precision	Recall	F1
T	39%	32%	34%
F	72%	80%	76%
Average	63%	66%	63%

As shown, the accuracy of the identical algorithms significantly decreased as follows:

(1) Precision dropped by (−21%), meaning that our measures have a positive effect on precision;
(2) Recall dropped by (−19%), meaning that our measures have a positive effect on recall;
(3) F1 dropped by (−22 %), meaning that our measures have a positive effect on F1.

Furthermore, we ran the test without using the non-topical features, only relying on topical features, including the semantic *NGD* measure. The results are shown in Table 7.

Table 7. The results as provided by the Random Forest algorithm, excluding any non-topical features (i.e., only using topical features).

	Precision	Recall	F1
T	53%	45%	50%
F	77%	80%	76%
Average	63%	66%	65%

It can be noted that there are noticeable improvements between the results shown in Tables 6 and 7, which highlight the importance of topical features, including *NGD* features.

7. Evaluation and Assessment

With an average F1 of 0.85, our method is successful in recognizing question paraphrasing and synonymy. The accuracy was enhanced due to the non-topical similarity metrics that were presented, particularly for the F-labeled questions. These findings were achieved without the use of a lexical dictionary, a semantic dictionary, or an ontological dictionary.

We infer from Table 5 that the T-labeled questions' precision is much lower than the F-labeled questions' precision. A possible explanation of this may be the fact that non-topical measures are extremely useful in deciding if two questions are distant (for instance, the proposed rules make it clear that "How" questions cannot be similar to "Who" questions). The identification of similar questions within the same scope, by comparison, needs more than just a resemblance in question types. It has been observed that some of the inaccuracies in T-labeled couples may be remedied by the use of a synonym lexicon (semantic network).

Our accuracy results are better than those achieved with similar Arabic [56] and non-Arabic experiments [57,58], as shown in Table 8. We acknowledge that the approaches below use different datasets and different performance metrics. However, Table 8 gives a clear indication that our approach has better or comparable results without using domain-dedicated

dictionaries, word embedding, or semantic networks, whereas all of the approaches below use word embedding and/or a semantic network. Furthermore, [56,59], in particular, ran experiments using datasets having similar sizes and similar performance metrics, and our system showed improved results.

Table 8. Comparison with the state-of-the-art systems for question similarity.

Name	Language	Task	Approach	Results
[56]	Arabic	Question similarity	Word embedding, and Deep learning	Accuracy, 58%, 77% on two different experiments on two different datasets
[59]	English	Question Similarity	Semantic networks: BabelNet, FrameNet	Average Precision, 76.7%
[49]	English	Question Similarity	Word embedding, machine translation	Accuracy, based on human judgment, 76%
[60]	Arabic	Question Similarity	Semantic networks: WordNet, and word embedding	Accuracy, based on human judgment, 75%

We think that making use of dictionaries, word embedding, a language model, and semantic networks that are domain-specific would enhance the outcomes even more, and this will be a primary focus of study in the future. However, in this experiment, we tried to prove the possibility of achieving good results without expensive and rich lexical resources.

8. Conclusions

Using topical and non-topical data and features, this research demonstrated a unique approach for calculating the degree to which Arabic questions are similar to one another. The topical techniques relied on string, lexical, and semantic similarity measures between the Arabic texts of the questions, whereas the non-topical approaches focused on the interrogative tools that were utilized by the Arabic questions. Both of the approaches showed effectiveness in accurately detecting similarity. For semantic similarity, we used Normalized Google Distance (*NGD*) as it does not require a textual corpus.

We presented the results of an experiment on a dataset of 2200 couples of Arabic questions collected from the Internet. Our proposed topical and non-topical features increased the accuracy of the results significantly in comparison to a simple model that utilizes baseline features. Our experiment results were closely comparable to those of other Arabic and non-Arabic experiments, despite not using a textual corpus or a lexical/semantic network. We believe that the results can be further improved with the utilization of a multi-domain Arabic lexical network, which will be part of our future work.

Funding: This research: including the APC was funded by the American University of Madaba.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this research paper is constructed by the authors and it is available at the following GitHub repository: https://github.com/Mohammad-Daoud-1/AQS (accessed on 1 June 2022).

Acknowledgments: The researcher would like to thank the American University of Madaba, in particular the Deanship of Scientific Research, for their support and help.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Vijaymeena, M.K.; Kavitha, K. A survey on similarity measures in text mining. *Mach. Learn. Appl. Int. J.* **2016**, *3*, 19–28.
2. Sayed, A.; al Muqrishi, A. An efficient and scalable Arabic semantic search engine based on a domain specific ontology and question answering. *Int. J. Web Inf. Syst.* **2016**, *12*, 242–262. [CrossRef]
3. Ye, X.; Shen, H.; Ma, X.; Bunescu, R.; Liu, C. From word embeddings to document similarities for improved information retrieval in software engineering. In Proceedings of the 38th International Conference on Software Engineering, Austin, TX, USA, 14–22 May 2016; pp. 404–415.

4. Wieting, J.; Berg-Kirkpatrick, T.; Gimpel, K.; Neubig, G. Beyond BLEU: Training Neural Machine Translation with Semantic Similarity. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; pp. 4344–4355.
5. Aggarwal, C.C.; Zhai, C.X. A survey of text clustering algorithms. In *Mining Text Data 9781461432*; Springer: Boston, MA, USA, 2012; pp. 77–128.
6. Seki, K.; Ikuta, Y.; Matsubayashi, Y. News-based business sentiment and its properties as an economic index. *Inf. Process. Manag.* **2022**, *59*, 102795. [CrossRef]
7. Guellil, I.; Adeel, A.; Azouaou, F.; Chennoufi, S.; Maafi, H.; Hamitouche, T. Detecting hate speech against politicians in Arabic community on social media. *Int. J. Web Inf. Syst.* **2020**, *16*, 295–313. [CrossRef]
8. Daoud, M.; Daoud, D. Sentimental event detection from Arabic tweets. *Int. J. Bus. Intell. Data Min.* **2020**, *17*, 471–492. [CrossRef]
9. Wang, Y.; Han, L.; Qian, Q.; Xia, J.; Li, J. Personalized Recommendation via Multi-dimensional Meta-paths Temporal Graph Probabilistic Spreading. *Inf. Process. Manag.* **2022**, *59*, 102787. [CrossRef]
10. Han, M.; Zhang, X.; Yuan, X.; Jiang, J.; Yun, W.; Gao, C. A survey on the techniques, applications, and performance of short text semantic similarity. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e5971. [CrossRef]
11. Levshina, N. Corpus-based typology: Applications, challenges and some solutions. *Linguist. Typology* **2021**, *26*, 129–160. [CrossRef]
12. Alwaneen, T.H.; Azmi, A.M.; Aboalsamh, H.A.; Cambria, E.; Hussain, A. Arabic question answering system: A survey. *Artif. Intell. Rev.* **2021**, *55*, 207–253. [CrossRef]
13. Shumanov, M.; Johnson, L. Making conversations with chatbots more personalized. *Comput. Human Behav.* **2021**, *117*, 106627. [CrossRef]
14. Gruber, T.R.; Brigham, C.D.; Keen, D.S.; Novick, G.; Phipps, B.S. *Using Context Information to Facilitate Processing of Commands in A Virtual Assistant*; United States Patent and Trademark Office: Washington, DC, USA, 2018.
15. Suhaili, S.M.; Salim, N.; Jambli, M.N. Service chatbots: A systematic review. *Expert Syst. Appl.* **2021**, *184*, 115461. [CrossRef]
16. Jurczyk, T.; Deshmane, A.; Choi, J.D. Analysis of Wikipedia-based Corpora for Question Answering. *arXiv* **2018**, arXiv:1801.02073.
17. Hamza, A.; En-Nahnahi, N.; Zidani, K.A.; Ouatik, S.E. An arabic question classification method based on new taxonomy and continuous distributed representation of words. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *33*, 218–224. [CrossRef]
18. Daoud, M. Building Arabic polarizerd lexicon from rated online customer reviews. In Proceedings of the 2017 International Conference on New Trends in Computing Sciences, ICTCS 2017, Amman, Jordan, 11–13 October 2017; pp. 241–246.
19. Silveira, C.R.; Santos, M.T.P.; Ribeiro, M.X. A flexible architecture for the pre-processing of solar satellite image time series data—The SETL architecture. *Int. J. Data Min. Model. Manag.* **2019**, *11*, 129–143.
20. Daoud, D.; Daoud, M. Extracting terminological relationships from historical patterns of social media terms. In *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; 9623 LNCS; Springer Science+Business Media: Berlin, Germany, 2018; pp. 218–229.
21. Grosan, C.; Abraham, A. Rule-Based Expert Systems. In *Intelligent Systems Reference Library*; Springer International Publishing: Cham, Switzerland, 2011; pp. 149–185.
22. Azad, H.K.; Deepak, A. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.* **2019**, *56*, 1698–1735. [CrossRef]
23. Prakoso, D.W.; Abdi, A.; Amrit, C. Short text similarity measurement methods: A review. *Soft Comput.* **2021**, *25*, 4699–4723. [CrossRef]
24. Tien, N.H.; Le, N.M.; Tomohiro, Y.; Tatsuya, I. Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. *Inf. Process. Manag.* **2019**, *56*, 102090. [CrossRef]
25. Ma, D.; Zhang, S.; Kong, F.; Cahyono, S.C. Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *662*, 052016.
26. Perumalla, S.R.; Eedi, H. Needleman–wunsch algorithm using multi-threading approach. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1090, pp. 289–300.
27. Abdeljaber, H.A. Automatic Arabic Short Answers Scoring Using Longest Common Subsequence and Arabic WordNet. *IEEE Access* **2021**, *9*, 76433–76445. [CrossRef]
28. Zhao, C.; Sahni, S. String correction using the Damerau-Levenshtein distance. *BMC Bioinform.* **2019**, *20*, 277. [CrossRef] [PubMed]
29. Wang, J.; Dong, Y. Measurement of Text Similarity: A Survey. *Information* **2020**, *11*, 421. [CrossRef]
30. Hamza, A.; Ouatik, S.e.; Zidani, K.A.; En-Nahnahi, N. Arabic duplicate questions detection based on contextual representation, class label matching, and structured self attention. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *34*, 3758–3765. [CrossRef]
31. Park, K.; Hong, J.S.; Kim, W. A Methodology Combining Cosine Similarity with Classifier for Text Classification. *Appl. Artif. Intell.* **2020**, *34*, 396–411. [CrossRef]
32. Wahyuningsih, T.; Henderi, H.; Winarno, W. Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient. *J. Appl. Data Sci.* **2021**, *2*, 45–54. [CrossRef]
33. Hasan, A.M.; Noor, N.M.; Rassem, T.H.; Noah, S.A.M.; Hasan, A.M. A Proposed Method Using the Semantic Similarity of WordNet 3.1 to Handle the Ambiguity to Apply in Social Media Text. *Lect. Notes Electr. Eng.* **2020**, *621*, 471–483.

34. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
35. Jatnika, D.; Bijaksana, M.A.; Suryani, A.A. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Comput. Sci.* **2019**, *157*, 160–167. [CrossRef]
36. Sangeetha, M.; Keerthika, P.; Devendran, K.; Sridhar, S.; Raagav, S.S.; Vigneshwar, T. Compute Query and Document Similarity using Explicit Semantic Analysis. In Proceedings of the 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 29–31 March 2022; pp. 761–766.
37. Baruah, N.; Gogoi, A.; Sarma, S.K.; Borah, R. Utilizing Corpus Statistics for Assamese Word Sense Disambiguation. In *Advances in Computing and Network Communications*; Springer: Singapore, 2021; Volume 736, pp. 271–283.
38. Ahmad, R.; Ahmad, T.; Pal, B.L.; Malviya, S. Approaches for Semantic Relatedness Computation for Big Data. In Proceedings of the 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, Sultanpur, India, 8–9 February 2019.
39. Tabassum, N.; Ahmad, T. Extracting Users' Explicit Preferences from Free-text using Second Order Co-occurrence PMI in Indian Matrimony. *Procedia Comput. Sci.* **2020**, *167*, 392–402. [CrossRef]
40. Kim, S.; Park, H.; Lee, J. Word2vec-based latent semantic analysis (W2V-LSA) for topic modeling: A study on blockchain technology trend analysis. *Expert Syst. Appl.* **2020**, *152*, 113401. [CrossRef]
41. Mittal, H.; Devi, M.S. Subjective Evaluation: A Comparison of Several Statistical Techniques. *Appl. Artif. Intell.* **2018**, *32*, 85–95. [CrossRef]
42. Prasetya, D.D.; Wibawa, A.P.; Hirashima, T. The performance of text similarity algorithms. *Int. J. Adv. Intell. Inform.* **2018**, *4*, 63–69. [CrossRef]
43. McCrae, J.P.; Rademaker, A.; Rudnicka, E.; Bond, F. English WordNet 2020: Improving and Extending a WordNet for English using an Open-Source Methodology. In Proceedings of the LREC 2020 Workshop on Multimodal Wordnets (MMW2020), Marseille, France, 11 May 2020; pp. 14–19.
44. Abdelali, A.; Darwish, K.; Durrani, N.; Mubarak, H. *Farasa: A Fast and Furious Segmenter for Arabic*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 11–16.
45. Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 1–14.
46. Nakov, P.; Màrquez, L.; Moschitti, A.; Magdy, W.; Mubarak, H.; Freihat, A.A.; Glass, J.; Randeree, B. SemEval-2016 Task 3: Community Question Answering. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 525–545.
47. Chen, X.; Zeynali, A.; Camargo, C.; Flöck, F.; Gaffney, D.; Grabowicz, P.; Hale, S.; Jurgens, D.; Samory, M. SemEval-2022 Task 8: Multilingual news article similarity. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Seattle, WA, USA, 14–15 July 2022; pp. 1094–1106.
48. Mihaylova, T.; Karadzhov, G.; Atanasova, P.; Baly, R.; Mohtarami, M.; Nakov, P. SemEval-2019 Task 8: Fact Checking in Community Question Answering Forums. In Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 860–869.
49. Nagoudi, E.M.B.; Ferrero, J.; Schwab, D.; Cherroun, H. Word Embedding-Based Approaches for Measuring Semantic Similarity of Arabic-English Sentences. *Commun. Comput. Inf. Sci.* **2018**, *782*, 19–33.
50. Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [CrossRef]
51. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
52. Alsultanny, Y.A. Machine Learning by Data Mining REPTree and M5P for Predicating Novel Information for PM10. *Cloud Comput. Data Sci.* **2020**, *1*, 40–48. [CrossRef]
53. Wang, F.; Li, Z.; He, F.; Wang, R.; Yu, W.; Nie, F. Feature Learning Viewpoint of Adaboost and a New Algorithm. *IEEE Access* **2019**, *7*, 149890–149899. [CrossRef]
54. Triayudi, A.; Widyarto, W.O. Comparison J48 And Naïve Bayes Methods in Educational Analysis. *J. Phys. Conf. Ser.* **2021**, *1933*, 012062. [CrossRef]
55. Kurani, A.; Doshi, P.; Vakharia, A.; Shah, M. A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting. *Ann. Data Sci.* **2021**, *2021*, 1–26. [CrossRef]
56. Einea, O.; Elnagar, A. Predicting semantic textual similarity of arabic question pairs using deep learning. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019.
57. Nakov, P.; Hoogeveen, D.; Màrquez, L.; Moschitti, A.; Mubarak, H.; Baldwin, T.; Verspoor, K. SemEval-2017 Task 3: Community Question Answering. *arXiv* **2017**, arXiv:1912.00730.
58. Galbraith, B.V.; Pratap, B.; Shank, D. Talla at SemEval-2017 Task 3: Identifying Similar Questions Through Paraphrase Detection. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017.

59. Franco-Salvador, M.; Kar, S.; Solorio, T.; Rosso, P. UH-PRHLT at SemEval-2016 Task 3: Combining Lexical and Semantic-based Features for Community Question Answering. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 814–821.
60. Wu, H.; Huang, H.; Jian, P.; Guo, Y.; Su, C. BIT at SemEval-2017 Task 1: Using Semantic Information Space to Evaluate Semantic Textual Similarity. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 77–84.