Novel Approach towards Arabic Question Similarity Detection

Mohammad Daoud
CS Department, Faculty of IT
American University of Madaba
Madaba, Jordan
m.daoud@aum.edu.jo

***Abstract*—In this paper we are addressing the automatic detection of Arabic question similarity, which is an essential issue in a variety of NLP/NLU applications such as question answering systems, virtual assistants, chatbots…etc. We are proposing and experimenting a rule-based approach that relies on lexical and semantic similarity between questions with the utilization of supervised learning algorithms. Our approach categorizes questions semantically according to their type and scope; this categorization is based on hypothetical rules that have been validated empirically, for example, a Timex Factoid question (a question asking about time) is less likely similar to an Enamex Factoid question (a question asking about a named entity). This article details the procedures of question pairs preprocessing, lexical analysis, feature extraction and selection and most importantly the similarity measures. According to the experiment we have conducted, our approach achieved promising precision and accuracy based on a test data of 1450 question pairs.**

Keywords—text similarity, question analysis, question similarity, semantic similarity, data science, Natural Language Processing.

I. INTRODUCTION

Finding similarity between various textual units (words, expressions, phrases, paragraphs …) is an important NLP task [1]. Many applications report significant improvements in their performance when a text similarity component is deployed, such as information retrieval [2], machine translation [3], text clustering [4], sentiment analysis [5]…etc. This task was tackled by researchers from different point of views, some methods assumes that two textual units are similar if they share subsequences of characters and words, for example, cosine similarity and Jaccard similarity [6] can be used as a simple similarity measure between phrases based on the common words between them. Semantic similarity tries to find logical similarity between texts even with the absence of the lexical similarity [7], for example, a semantic network or a corpus can be used to determine the degree of similarity between two words or expressions even if the text seems different in terms of its characters and words [8].

Similarity between questions is an interesting task that can be very helpful for a series of applications such as question answering systems [9], virtual assistants [10], chatbots [11]…etc. It can be considered as a sub problem of text similarity. The challenge here is that questions are difficult to be processed and has short to no textual context. Besides, questions are paraphrased more often that other utterances [12].

Arabic questions similarity is even more challenging, because Arabic is a pi-language (poorly informatized language) [13] [14] and gaining semantic information from its corpus is difficult. Few research attempts have addressed Arabic question similarity where mediocre results have been achieved (when compared to other resourceful languages) [15].

With the absence or the scarceness of relevant semantic corpus for Arabic, a rule-based system for categorizing questions can be used [16]. In this paper we are seeking a hybrid approach that utilizes supervised learning and hypothetical rules to find similarity and to detect paraphrasing.

Many researchers focus only on corpus data-driven approaches to cluster, classify and map words and phrases [7] [17]. We believe that this is an essential part of the similarity detection task. However, in the context of question similarity, certain rules can be set to improve the understanding of the questions and to relate them accordingly, for example, these two questions are distanced even though they have high string similarity, high term similarity, and high semantic similarity, simply because the first one asks about the time and the second one asks about a location. *Q1 = "Arabic: متى وقعت غزوة بدر؟ - English: When did the Battle of Badr take place?" Q2 = "Arabic: اين وقعت غزوة بدر؟ - English: Where did the Battle of Badr take place?"*. In this paper we are forming a framework to understand the Arabic questions and to use this in improving question similarity.

This paper is organized as follows: the next section lists and compares the most relevant related work. After that, in section three we introduce our approach in question comparison and analysis. In section four we detail the aspects of the data set we are using for the experiment, and the preprocessing method. And then section five shows the experiment and its results, while section six evaluates and assesses our method. And finally, we draw some conclusions, future work and possible applications.

II. RELATED WORK

Similarity between phrases can be approached through textual (String) similarity and semantic similarity. Question similarity, which is the focus of this paper, is a sub problem of phrasal similarity. Therefore, this section will address

978-1-7281-2882-5/19/$31.00 ©2019 IEEE

phrasal similarity in general and then will discuss attempts of Arabic question similarity detection.

Textual similarity [18] relies on the string representation of phrases. And therefore, simply, two phrases are similar if they have similar strings. There are two main approaches in string similarity; the first one treats the phrase as a sequence of characters [19] and the second one treats phrases as lexical units glued with a syntax [20]. Longest common subsequence [21], Jaro [22], Damerau-Levenshtein [23], and Needleman-Wunsch [24] are considered amongst the most frequently used character-based similarity algorithms. While Block Distance, Cosine similarity [25], Dice's coefficient [26], Euclidean distance (L2), and Jaccard similarity [6] are well known algorithms for lexical-based similarity [27]. The advantage of these two approaches is that they are simple, and effective for short phrases that belong to the same domain, where there is limited word ambiguity.

Semantic similarity can be effective to address word ambiguity [17]. It tries to map different lexical units based on their meaning distance, regardless of their string distance. Most of the semantic similarity algorithms rely on large corpus to extract additional information about the constructs of the phrase. For example, finding the similar words based on their frequent colocation. The following algorithms and methods are considered as corpus semantic similarity algorithms: Hyperspace Analogue to Language (HAL) [28], Latent Semantic Analysis (LSA) [29], Generalized Latent Semantic Analysis (GLSA) [30], Explicit Semantic Analysis (ESA) [31], Pointwise Mutual Information - Information Retrieval (PMI-IR) [32], Second-order co-occurrence pointwise mutual information (SCO-PMI) [33], Normalized Google Distance (NGD) [34] and Extracting DIStributionally similar words using COoccurrences (DISCO) [35]. These algorithms are effective only with the availability of large and clean corpus, and they assume relatedness based on the textual colocations.

Usually, a semantic network is augmented to the semantic similarity engine such as Wordnet [36]. In fact many researchers are using Wordnet heavily to measure the distances between words and phrases which can be considered as an independent semantic similarity measure. Which is effective for resourceful languages such as English (English Wordnet has 155 327 words organized in 175 979 synsets).

We are proposing a hybrid approach that utilizes string similarity and semantic similarity but without demanding huge resources, which is still considered a problem for languages such as Arabic.

III. Question Comparison

In this paper we are introducing a novel method to determine similarity between two Arabic questions. Our algorithm employs textual and semantic similarity. This section details our approach, starting with the main algorithm, preprocessing, feature generation and question scope analysis.

A. *Lexical and Semantic Similarity*

To compare between two questions, we generate a list of features for every couple. Algorithm 1 receives q1 and q2 and utilizes certain similarity measures to produce a list of features that belong to the couple (q1 - q2).

```
Algorithm 1
FindSimilarityFeatures( Couples of questions C )
// start of Algorithm 1
    For each couple cx ( q1 , q2 ) in C
            nq1 = Normalize ( q1 )
            nq2 = Normalize ( q2 )
            nqq1 = QuestionNormalization ( nq1 )
            nqq2 = QuestionNormalization ( nq2 )
            bowq1 = BOW ( nqq1 )
            bowq2 = BOW ( nqq2 )
            nerq1 = NER ( q1 )
            nerq2 = NER ( q2 )
            Posq1 = pos ( nqq1 )
            Posq2 = pos ( nqq2 )
    F [ x ] [ ] =     { lcs ( nq1 , nq2 ) ,
            cos ( bowq1 , bowq2 ) ,
            jaccard ( bowq1 , bowq2 ) ,
            euc ( bowq1 , bowq2 ) ,
            jaccard ( nerq1 , nerq2 ) ,
            cos ( nerq1 , nerq2 ) ,
            jaccard ( posq1 , posq2 ) ,
            cos ( posq1 , posq2 ) ,
            Startsim ( bowq1 , bowq2 ) ,
            Endsim ( bowq1 , bowq2 ) ,
            QWsim ( bowq1 , bowq2 ) }
Return F
// end of Algorithm 1
```

The algorithm starts by normalizing the Arabic text of q1 and q2. Then special question normalization is done as shown in Algorithm 2, where nonstandard question words and expressions are detected and replaced by standard words. This will eliminate unnecessary variations and will result in more accurate similarity measures. Algorithm 2 is equipped with a list of nonstandard question words and their standard equivalences. The list is sorted according to the length of the nonstandard question words, so that the algorithm will make longest match detection.

```
Algorithm 2
QuestionNormalization ( q )
// start of algorithm 2
    Read table1 ( n , s ) [ ]
    //table1 (n = non standard question words,
    //standard question form)
    // table1 is sorted according to the numbers of
    //words of n descending order
    For each t ( n , s ) in table1 [ ]
            q.replace ( n , s )
Return q
// end of Algorithm 2
```

After question normalization, the similarity will be measured between the following (1) bag of words from the normalized q1 and q2 (2) the named entity in q1 and q2 (3) q1 and q2 after part of speech tagging. For named entity recognition (NER) and for part of speech (POS) analysis we use [37].

Algorithm 3
Startsim(q1 [ ] , q2 [ ])
// start of algorithm 3
If q1 [0] == q2 [0] and q1[1] = = q2 [1]
Return 1
Else if q1 [0] == q2 [0]
Return 0
Else
Return – 1
// end of algorithm 3

Algorithm 1 generates the following features:

1. Longest common subsequence between the normalized q1 and q2
2. Cosine similarity between the normalized BOW of q1 and q2
3. Jaccard similarity between the normalized BOW of q1 and q2
4. Euclidian distance between the normalized BOW of q1 and q2
5. Jaccard similarity between the named entity of q1 and q2
6. Cosine similarity between the named entity of q1 and q2
7. Jaccard similarity between the part of speech analysis for q1 and q2
8. Cosine similarity between the part of speech analysis for q1 and q2
9. Start similarity which is described in algorithm 3
10. End similarity which is described in algorithm 4
11. Question word similarity which is described in algorithm 5

Algorithm 3 returns 1 if the 2 starting question words in q1 and q2 are the same. It returns 0 if only the first word in q1 is equivalent to the first word in q2. And it returns -1 if the first and the second words in q1 are not the same as the words in q2.

Algorithm 4
Endsim(q1 [ ] , q2 [ ])
// start of algorithm 4
If q1 [q1.length – 1] == q2 [q2.length – 1]
and q1 [q1.length – 2] == q2[q2.length – 2]
Return 1
Else if q1 [q1.length – 1] == q2 [q2.length – 1]
Return 0
Else
Return – 1
// end of algorithm 4

Algorithm 4 returns 1 if the last 2 words in q1 and q2 are the same. It returns 0 if the last word in q1 is equivalent to the last word in q2. And it returns -1 if the last two words in q1 are not the same as the words in q2. The idea behind the feature generated by algorithm 4 is simple, some couple might produce high textual similarity, and however, the dissimilarity of the last 1 or two words might alter the focus of the questions completely.

Algorithm 5
QWsim(q1 [ ] , q2 [ ])
// start of algorithm 5
qw1 = Getquestionword (q1)
qw2 = Getquestionword(q2)
if qw1 and qw2 belong to same scope
Return 1
else if qw1 and qw2 belong to related scopes
Return 0
else
Return – 1
// end of algorithm 5

Algorithm 5 calculates similarity based on question type, it returns 1 if q1 and q2 are of the same type and scope. It returns 0 if they have related scopes, and it returns -1 if they have completely different scopes. Getquestionword is afunction that detects the question word(s) that has been in thequestion. Next section discusses question scopes analysis in details.

B. *Semantic similarity*

Table 1 suggest a categorization of the main scopes of Arabic questions, as we can see; each scope is categorized by the possible question. The answer of a TimexF question would be a time or date. While the answer of a LocF question is a location. Semantically the two question will most likely get two different answers and therefore, they have a semantic distance, even if the two questions are lexically similar.

TABLE 1. Scopes of Arabic questions

ID	Scope	Question words	Paraphrased words
TimexF	Time - Factoid	متى, ايان “When”	في اي وقت “in what time” في اي سنة “in what year” ما هو تاريخ “what is the date”
LocF	Location - Factoid	أين Where	ما موقع “What is the location” في اي مدينة “in what city” “in which country” في اي دولة
NVF	Numeric value - Factoid	كم How many How Much	ما طول “what is the length” ما هي المسافة “what is the distance” ما عرض “what is the width”
NEF	Named Entity - Factoid	لمن Whose	الى من “for whom” من هو “Who is”

			لاي “For whom”
NED	Named Entity - Definition	من, ما What	ما تعريف “what is the difinition” من هو “Who is”
M	Method	كيف How	ما هي طريقة “What is the method” ما هو وصفة “What is the recipe” ما الخطوات “What are the steps”
P	Purpose	لماذا Why	ما هو السبب “ what is the reason” ما المسبب “What causes”
C	Cause	ماذا What	ما الذي “What”
L	List	اذكر, عدد List	
YN	Yes/No	هل Is/was/are…	ء “Question Hamza”

We seek to give a similarity measure for a couple of Arabic questions based on the scope of their interrogative word (question word). We use empirical and hypothetical approaches to establish the needed rules.

It is intuitive that a method question that starts with “كيف - How” will be dissimilar to a factoid timex question that starts with “متى - when” and based on that we can hypothesize the following rule:

If q1.scope = M and q2.scope = TimexF then qw1 = -1

This hypothetical rule can be confirmed empirically by an experiment. In the same way we assumed that if the scope of the two questions is the same then they have a similarity measure of 1.

We found out through the experiment that some of the scopes have unconfirmed similarity such as NEF – NED, and P – M. Therefore, such occurrence would result in a 0 similarity measure.

IV. Data preparation

For experimentation, we have selected 300 Arabic questions from the Frequently Asked pages of various United Nation’s organizations. And we have randomly selected 300 interrelated casual Arabic questions from ejaaba.com. We used these 600 questions to randomly generate 1450 couples. Each couple was given a YES or NO label, to indicate the similarity of the two questions. 419 couples were labeled with a YES, and 1031 couples were labeled as NO. Because it was difficult to find YES-labeled questions in the randomly generated couples, we used paraphrasing to generate half of the YES-labeled couples and we used the same technique with 100 NO-labeled questions.

The 1450 couples were normalized (Arabic and question normalization) and then used to generate the features described in section III.

The distribution of the scopes of the 600 unique questions was as shown in table 2.

TABLE 2. The distribution of the scopes of the 600 unique questions

Scope	Number of questions
Time - Factoid	88
Location - Factoid	79
Numeric value - Factoid	69
Named Entity - Factoid	27
Named Entity - Definition	55
Method	78
Purpose	48
Cause	45
List	19
Yes/No	92

V. Experiment

We used several classification algorithms provided by WEKA 3.8 [38] on the generated data set. Random Forests [39] with 10 folds cross validation has produced the best results amongst other classifiers that we have tested in terms of precision, recall and f-Measures.

Table 3 shows the results reported from Random Forests Classifier.

TABLE 3. Results reported by Random Forests Algorithm, with our proposed features

	Precision	Recall	F-measures
Yes	0.82	0.59	0.69
No	0.85	0.95	0.90
Weighted Avg.	0.84	0.85	0.84

To evaluate our novel approach we ran the test after removing our special features (End similarity, Start Similarity, Question Word Similarity), and therefore the remaining features were simply based on cosine similarity, jaccard similarity, Euclidean distance and Longest Common Subsequence. Table 4 shows results for the same test but without our features.

TABLE 4. Results reported by Random Forests Algorithm, without our proposed features

	Precision	Recall	F-measures
Yes	0.40	0.32	0.35
No	0.74	0.80	0.77
Weighted Avg.	0.64	0.66	0.65

As you can see there is a significant drop in accuracy for the same algorithms in terms of precision (-0.2), recall (-0.19) and F-measures (-0.19).

VI. Evaluation and Assessment

Our system can detect question paraphrasing and synonymy with an overall precision of 0.85. The proposed question type similarity increased the accuracy, especially for NO-labeled questions. This was achieved without using a lexical or semantic dictionary.

From table 3 we notice that the accuracy of the YES-labeled questions is behind the accuracy of the NO-Labeled questions and that can be due to the fact that question type similarity was very effective in determining if two questions are dissimilar (for example, “When” questions can’t be similar to “Where” questions, and that can be easily determined). However, determining similar questions within the same scope needs more than question type similarity. We noticed that some of the YES-Labeled errors could be avoided by a simple synonymy lexicon.

Our accuracy results are comparable with similar experiments, even those that were performed on resourceful languages such as English [40] [41].

We believe that utilizing a domain dedicated lexicon can improve the results even more, and that is definitely a future research focus.

VII. Conclusion

We have presented a novel approach to detect similarity between Arabic questions. Our rule based similarity algorithm showed effectiveness according to the experiment we have conducted, despite its limited dependency on a lexical resource. String based similarity and lexical based similarity can be used as a base for our algorithm, but they have narrow capabilities and thus our proposed similarity measures presented in this paper has improved accuracy and precision. The results obtained by the experiment were comparable to similar experiments in the English language, which is significant considering that English is a resource rich language if compared to Arabic. We anticipate that the result will be improved furthermore with the help of a carefully constructed multi domain Arabic lexicon. And this is part of our future work.

References

[1] M. K. Vijaymeena and K. Kavitha, “A survey on similarity measures in text mining,” *Mach. Learn. Appl. An Int. J.*, vol. 3, no. 2, pp. 19–28, 2016.

[2] X. Ye, H. Shen, X. Ma, R. Bunescu, and C. Liu, “From word embeddings to document similarities for improved information retrieval in software engineering,” in *Proceedings of the 38th international conference on software engineering*, 2016, pp. 404–415.

[3] M. Simard, N. Ueffing, P. Isabelle, and R. Kuhn, “Rule-based translation with statistical phrase-based post-editing,” in *dl.acm.org*, 2007, pp. 203–206.

[4] C. C. Aggarwal and C. X. Zhai, “A survey of text clustering algorithms,” in *Mining Text Data*, vol. 9781461432, Boston, MA: Springer US, 2012, pp. 77–128.

[5] B. Pang and L. Lee, *Opinion Mining and Sentiment Analysis: Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2. 2008.

[6] A. Huang, “Similarity measures for text document clustering,” in *New Zealand Computer Science Research Student Conference, NZCSRSC 2008 - Proceedings*, 2008, pp. 49–56.

[7] A. Islam, “Semantic text similarity using corpus-based word similarity and string similarity,” *ACM Trans. Knowl. Discov.*

[8] M. Steyvers and J. B. Tenenbaum, “The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth,” *Cogn. Sci.*, vol. 29, no. 1, pp. 41–78, 2005.

[9] J. Weston *et al.*, “Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks,” *arxiv.org*, 2015.

[10] T. R. Gruber, C. D. Brigham, D. S. Keen, G. Novick, and B. S. Phipps, “Using Context Information to Facilitate Processing of Commands in A Virtual Assistant,” *Washington, DC U.S. Pat. Trademark Off.*, 2018.

[11] N. M. Radziwill and M. C. Benton, “Evaluating Quality of Chatbots and Intelligent Conversational Agents,” Apr. 2017.

[12] T. Jurczyk, A. Deshmane, and J. D. Choi, “Analysis of Wikipedia-based Corpora for Question Answering,” Jan. 2018.

[13] M. Daoud, “Building Arabic polarizerd lexicon from rated online customer reviews,” in *Proceedings - 2017 International Conference on New Trends in Computing Sciences, ICTCS 2017*, 2018, vol. 2018-Janua, pp. 241–246.

[14] C. R. Silveira, M. T. P. Santos, and M. X. Ribeiro, “A flexible architecture for the pre-processing of solar satellite image time series data - The SETL architecture,” *Int. J. Data Mining, Model. Manag.*, vol. 11, no. 2, pp. 129–143, 2019.

[15] A. Hamza, N. En-Nahnahi, K. A. Zidani, and S. El Alaoui Ouatik, “An arabic question classification method based on new taxonomy and continuous distributed representation of words,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2019.

[16] C. Grosan and A. Abraham, “Rule-Based Expert Systems,” 2011, pp. 149–185.

[17] A. Prior and M. Geffet, “Word Association Strength, Mutual Information and Semantic Similarity,” in *EuroCogSci 2003*, 2003.

[18] J. Lu, C. Lin, W. Wang, C. Li, and H. Wang, “String similarity measures and joins with synonyms,” in *Proceedings of the 2013 international conference on Management of data - SIGMOD ’13*, 2013, p. 373.

[19] G. Navarro and Gonzalo, “A guided tour to approximate string matching,” *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31–88, Mar. 2001.

[20] P. Gamallo, C. Gasperin, A. Agustini, and G. P. Lopes, "Syntactic-Based Methods for Measuring Word Similarity," Springer, Berlin, Heidelberg, 2001, pp. 116–125.

[21] A. Apostolico and C. Guerra, "The longest common subsequence problem revisited," *Algorithmica*, vol. 2, no. 1–4, pp. 315–336, Nov. 1987.

[22] P. Angeles and A. Espino-gamez, "Comparison of methods Hamming Distance , Jaro , and Monge-Elkan," *DBKDA 2015 Seventh Int. Conf. Adv. Databases, Knowledge, Data Appl.*, no. c, pp. 63–69, 2015.

[23] F. Miller, A. Vandome, and J. McBrewster, "distance: Information theory, computer science, string (computer science), string metric, damerau? Levenshtein distance, spell checker, hamming distance," 2009.

[24] V. Liki, "The Needleman-Wunsch algorithm for sequence alignment 7th Melbourne Bioinformatics Course," *cs.sjsu.edu*, pp. 1–46.

[25] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the National Conference on Artificial Intelligence*, 2006, vol. 1, pp. 775–780.

[26] N. Oco, L. R. Syliongka, R. E. Roxas, and J. Ilao, "Dice's coefficient on trigram profiles as metric for language similarity," in *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, 2013, pp. 1–4.

[27] D. Daoud and M. Daoud, "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)*, 2018, vol. 9623 LNCS, pp. 218–229.

[28] L. Azzopardi, M. Girolami, and M. Crowe, "Probabilistic hyperspace analogue to language," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '05*, 2005, p. 575.

[29] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*, 1999, vol. 51, no. 2, pp. 50–57.

[30] M. Monjurul Islam and A. S. M. Latiful Hoque, "Automated essay scoring using Generalized Latent Semantic Analysis," in *2010 13th International Conference on Computer and Information Technology (ICCIT)*, 2010, pp. 358–363.

[31] O. Egozi, S. Markovitch, and E. Gabrilovich, "Concept-Based Information Retrieval Using Explicit Semantic Analysis," *ACM Trans. Inf. Syst.*, vol. 29, no. 2, pp. 1–34, Apr. 2011.

[32] G. Bouma, "Normalized (Pointwise) Mutual Information in Collocation Extraction."

[33] M. A. Islam and D. Inkpen, "Second Order Co-occurrence PMI for determining the semantic similarity of words," in *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*, 2006, pp. 1033–1038.

[34] R. L. Cilibrasi and P. M. B. Vitanyi, "The Google Similarity Distance," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 370–383, Mar. 2007.

[35] P. Kolb, "Disco: A multilingual database of distributionally similar words," *Nat. Lang. Process.*, no. 2003, pp. 37–44, 2008.

[36] G. A. Miller, "WordNet: A Lexical Database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[37] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A Fast and Furious Segmenter for Arabic."

[38] E. Frank *et al.*, "Weka-A Machine Learning Workbench for Data Mining," in *Data Mining and Knowledge Discovery Handbook*, Boston, MA: Springer US, 2009, pp. 1269–1277.

[39] L. Breiman, "Random forests," *Mach. Learn.*, pp. 5–32, 2001.

[40] P. Nakov *et al.*, "SemEval-2017 Task 3: Community Question Answering."

[41] B. V Galbraith, B. Pratap, and D. Shank, "Talla at SemEval-2017 Task 3: Identifying Similar Questions Through Paraphrase Detection."