ST NSURL 2019 Shared Task: Semantic Question Similarity in Arabic

Mohamed Lichouri, Mourad Abbas, Besma Benaziz
Computational Linguistics Dept.,CRSTDLA
Algeria
m.lichouri@crstdla.dz
m.abbas@crstdla.dz
b.benaziz@crstdla.dz

Abed Alhakim Freihat
University of Trento
Italy
abed.freihat@unitn.it

Abstract

In this paper, we describe the solution that we propose for the shared task NSURL 2019 Semantic Question Similarity in Arabic. The proposed solution combines three approaches: lexical, statistical, and neural. The lexical approach is based on similarity measures. The statistical approach utilizes a set of binary classifiers. The neural approach uses a Siamese Deep Neural Network Model.

1 Introduction

The task in NSURL 2019 Semantic Question Similarity in Arabic shared task (Seelawi et al., 2019) is to predict the semantic similarity between questions: For a given question pair $< q_1, q_2 >$, identify if the questions q_1 and q_2 have the same meaning or not.

A question similarity system is an important component that contributes to good question answering portal. This component enables users to find answers to previously asked questions similar to their own before posting new questions.

Many Question similarity approaches have been already proposed for English (Nakov et al., 2016) and other European languages such as Spanish, French or Italian (Buscaldi et al., 2010).

For the Arabic language, there are also some question similarity proposals (Abouenour et al., 2010). were Such approaches do not give a general solution to the problem of question semantic similarity due to some limitations these systems have. For example, QARAB (Hammo et al., 2002) system does not take into consideration the understanding of the content of the question at a semantic level. AQAS (Mohammed et al., 1993) system is designed fro structured texts only. ArabiQA (Benajiba et al., 2007) and QASAL (Brini et al., 2009) Systems target factoid questions only.

		# Sentences	# Words
Train	**Ques1**	11,995	68,608
	Ques2	11,995	64,039
	QuesPairs	11,995	64,039
Test	**Ques1**	3,715	21,248
	Ques2	3,715	19,682
	QuesPairs	3,715	19,682
Total	**QuesPairs**	15,710	83,721

Table 1: Statistics of the used dataset.

The proposed system in this paper combines three approaches: lexical, statistical and neural. In the lexical approach, we use a set of text similarity measures from the text distance tools. In the statistical approach, we deploy a set of classifiers. In the neural approach, we apply a Siamese Deep Neural Network Model. We also use additional features such as punctuation and stop word filtering, normalization, stemming, and POS-tagging to enhance the final results.

The rest of the paper is organized as follows; we describe our data in Section 2 and the proposed system in Section 3. We report our experiments and results in 4 and conclude with conclusion and suggestions for future research in 5.

2 Dataset

In this work, we used the NSURL Task8(Seelawi et al., 2019) data set provided by the Mawdoo3 Team. The training data is composed of 11995 sentences. The size of the test sets is 200 questions pairs. The questions are short, ranging from 4 to 15 words each. Each sentence is annotated with the speaker dialect. In table 1, we provide some statistics on the used corpora.

3 System

The presentation of our proposed system is shown in figure 1.

In the following, we summarize the approach:

1. In parallel, run the three approaches (lexical, statistical, and neural).
2. Select the three best configurations that achieved the best performance.
3. In the third step, apply a combination of features which will give us the best model for each approach.
4. In the last step, combine two of the three models 2by2.This enables us to have a lexical-statistical combination approach and a neural approach.

3.1 Features extraction

3.1.1 ngrams features

The first features that we considered to deal with the problem of Semantic Question Similarity in Arabic, were the word and character n-grams features used in previous work such as (Salameh et al., 2018; Lichouri et al., 2018), where we added another feature which is the character-word_boundary (char_wb). In the following, we present a description of the three adopted features.

- **[Word n-grams:]** We extract n-gram word from 1st to 5th.
- **[Char n-grams:]** The character 1st to 5th grams are used as features.
- **[Char_wb n-grams:]** This feature creates character n-grams only from text inside word boundaries; n-grams at the edges of words are padded with space.

3.1.2 Additional Features

The features considered are obtained by applying three processes, either simultaneously or individually. These process are: Punctuation removable, Stop-word filtering, Normalization Process, Stemmer Process and a PosTagger Process. To deal with the last three process, we first defined our own normalizer function, then used the ISRIStemmer NLTK tool[1] for the second, whereas for last we used the NLTK postagger[2].

3.2 Proposed Approches

3.2.1 Lexical Approach

This approach is based on a set of text distance measures from the textdistance tools[3]. From a set of measures proposed by this tools we opted to choose one measure per category, namely: Hamming Distance, Mlipns Distance, Levenshtein Distance, Damerau Levenshtein Distance, Jaro Distance, Strcmp95 Distance, Needleman Wunsch Distance, Gotoh Distance, and the Smith Waterman Distance.

3.2.2 Statistical Approach

Based on a set of classifiers using the scikit-learn library (Pedregosa et al., 2011), namely: Linear Support Vector Classification (LSVC), Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB), Logistic Regression (LGR), Stochastic Gradient Descent (SGD), Perceptron (PRP) and the Passive Aggressive (PAG), a statistical approach was proposed. Where we will consider the semantically similarity between questions as a binary classification problem with two classes: similar (1) or non similar (0).

3.2.3 Neural Approach

In this approach, we will consider a Text Classification Methods using a Siamese Deep Neural Network [4]. While using this script, we adopted for multiple configurations by varying the default setup: EMBEDDING_DIM = 50, MAX_SEQUENCE_LENGTH = 10, VALIDATION_SPLIT = 0.1, RATE_DROP_LSTM = 0.17, RATE_DROP_DENSE = 0.25, NUMBER_LSTM = 50, NUMBER_DENSE_UNITS = 50, ACTIVATION_FUNCTION = ’relu’.

4 Results

As shown in figure 1, the first step to be conducted is to experiment with the three approaches in parallel. For the first approach, which is the lexical approach, we conducted a similarity measure study, by calculating the distance between the two questions by using several metrics while considering a range of threshold values between 10% to 100%. The best results are presented in table 2.

The three best results obtained by this approach are by the following measures: Smith Water-

[1] https://kite.com/python/docs/nltk.stem.ISRIStemmer

[2] https://www.nltk.org/_modules/nltk/tag.htm

[3] https://pypi.org/project/textdistance/

[4] https://github.com/amansrivastava17/lstm-siamese-text-similarity

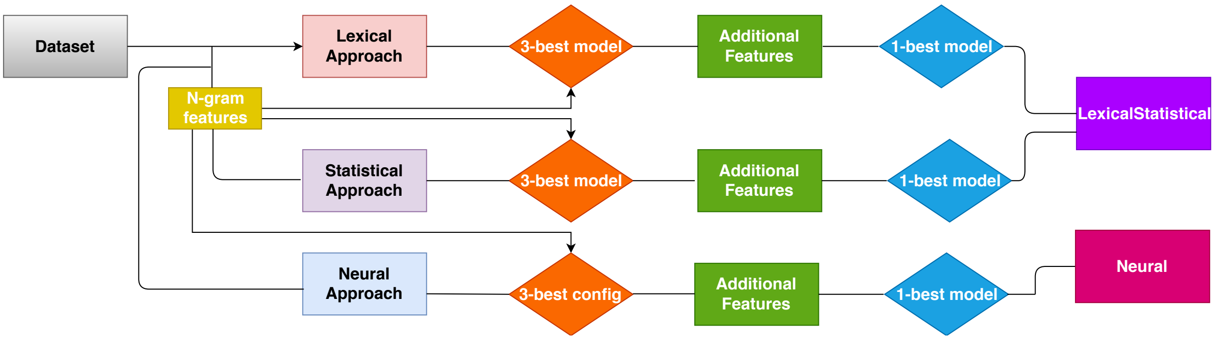

Figure 1: A Semantic Question Similarity System for Arabic Language

Similarity Measures	Threshold (%)	Score (%)
Hamming Distance	90	47.95
Mlipns Distance	90	38.82
Levenshtein Distance	40	**68.28**
Damerau Levenshtein Distance	40	67.91
Jaro Distance	30	64.20
Strcmp95 Distance	30	63.27
Needleman Wunsch Distance	50	**67.93**
Gotoh Distance	40	67.91
Smith Waterman Distance	30	**69.77**

Table 2: Best results obtained by the different measures while varying the threshold.

	MNB	BNB	LSVC	LGR	PRP	PAG	SGD
Unigram	63.11	63.98	70.86	69.70	66.43	68.81	70.87
Bigram	63.35	62.64	72.79	72.01	70.51	72.04	73.48
Trigram	62.41	60.46	73.94	72.55	69.75	72.50	73.98
4-grams	61.57	57.99	**74.25**	72.96	71.04	73.07	74.03
5-grams	60.70	56.34	74.11	73.11	71.17	**73.36**	**74.73**

Table 3: Results obtained by the used classifiers in term of F1-score while varying the number of grams n with the word feature

man distance, Levenshtein distance and Needleman Wunsch Distance. The best score obtained is around 69.77% with a threshold of 30. For the second approach, the statistical one, we used the n-grams word and char features, with a range of n from 1 to 5. The results obtained while applying these features with the aforementioned classifiers are presented in tables 3 and 4.

It should be noted, that with this approach; the three best results were obtained by the LSVC, PAG and SGD while using the word (4/5)-grams feature. The f1-score obtained range between $\tilde{7}2\%$ and $\tilde{7}5\%$.

	MNB	BNB	LSVC	LGR	PRP	PAG	SGD
Unigram	44.48	64.72	67.60	66.59	57.57	54.60	66.38
Bigram	61.70	66.45	71.39	69.76	63.98	64.79	69.77
Trigram	**63.18**	**66.47**	73.03	70.98	68.13	66.86	71.28
4-grams	62.30	65.97	73.43	70.94	**69.51**	69.84	72.51
5-grams	61.68	65.37	**74.03**	**71.06**	69.48	**71.70**	**72.95**

Table 4: Results obtained by the used classifiers in term of F1-score while varying the number of grams n with the char feature

	max_length	function	threshold	F1-Score
config1	10	softmax	0.5	79.89
config2	10	softmax	0.55	79.89
config3	15	softmax	0.55	79.11

Table 5: Results obtained by the three best configurations in term of F1-score while varying the different parameters

		Development Score	
Approach	**Models**	**Public (30%)**	**Private (70%)**
Lexical	***NW***	70.37	68.08
	Lv	69.12	68.35
	SW	72.26	69.78
Statistical	***LSVC***	75.31	75.62
	SGD	75.13	74.77
	PAG	74.59	73.58
Neural	***Config1***	81.23	80.54
	Config2	82.31	79.77
	Config3	82.58	82.58

Table 6: Development results obtained by the three best models for each approach in the first step.

Whereas for third approach, based on the Siamese DNN and as mentioned in the description of the neural approach, we have experimented with multiple combination of values for each parameters and thus noted the three best configurations, which we presented in table 5. We can note that there is a net amelioration of the F1-score against the two previous approach with an amelioration of +5 point.

Before presenting the results that we obtained in the second step, we will report the development accuracy that we obtained after submitting the three best model for the three approach to the kaggle shared task website in table 6.

From the table 6, we can see that the best results

	Baseline	Features								
		P	S	N	St	Pos	PSNPos	PSNSt	PSNStPos
SW	69.77(30)	69.86(30)	70.62(30)	69.86(30)	69.77(30)	68.71(30)	**71.83(30)**	71.23(30)	**71.83(30)**
Lv	68.28(40)	68.46(40)	69.50(30)	68.17(40)	68.28(40)	66.90(30)	68.36(30)	70.65(40)	68.36(30)
NW	67.91(50)	68.30(50)	69.15(50)	67.93(50)	67.91(50)	66.78(30)	68.04(50)	70.01(50)	68.04(50)
LSVC(4)	74.25	73.94	69.58	72.86	73.79	**75.69**	74.16	69.60	74.37
SGD(5)	74.73	74.31	69.05	74.13	74.1	74.49	74.92	68.65	74.49
PAG(5)	73.36	72.84	68.62	74.01	72.96	73.53	73.37	68.02	73.28
DNN	79.89	77.29	75.65	76.66	74.76	**80.55**	74.46	74.66	75.82

Table 7: Comparison of the impact of the preprocessing step on the results obtained by the best models in the baseline system, in accordance to the three proposed approach. For the first parts of the table(lexical approach), we noted the threshold that gave us the best results in brackets.

			Development Score	
Approach	**Models**	**Features**	**Public (30%)**	**Private (70%)**
Lexical	**SmithWater**	**PNSStPos**	73.16	71.55
Statistical	*LSVC*	**PosTagger**	81.68	79.04
Neural	*Config1*	**PosTagger**	59.97	62.11

Table 8: Development results obtained by the 1-best models for each approach in the second step.

	Development Score	
Approach	**Public (30%)**	**Private (70%)**
1st Best Model	83.57	82.69
2nd Best Model	82.58	80.50
Benchmark	71.99	71.43

Table 9: Comparison of our best model performance against the benchmark.

are obtained by the neural approach in both test dataset (30% and 70%) with an average f1-score of more than 80%.

We will now, present the results obtained in the second step, where we applied a set of additional features. This step will permit us to select the best model for each approach. The table 7, present the gotten results.

By applying some additional features namely: Punctuation removal, Stopwords filter, Normalizer process, Stemmer process and PosTagger process, individually or sequentially, we can note a net amelioration of results for all the three approach by +̃2, +̃2 and +̃1 for the lexical, statistical and neural approaches, respectively.

When looking at the table 7, we can infer that the best model for each approach is as follows: the Smith Waterman distance for the lexical approach, the LSVC classifier for the statistical approach and the DNN+Postagger for the neural approach.

As we did before, we have re-submitted the best model for each approach to the kaggle to have the score with the test dataset. The gotten results are demonstrated in table 8.

We can note that despite the neural approach has scored the best score of 80.55% in the training phase, it could not well generalize on the test data, where it yielded 59.97% and 62.11% for both the public and private set. For the third step, we have compared the performance of a combination between the lexical and statistical approaches against the neural approach, which have given us two model: lexical+statistical and neural.

We started with the statistical approach, where we have opted to add a combination of features, which has given rise to a new features. This new feature contains:

- A 5-grams word feature.
- A 3-grams char feature.
- A 3-grams char_wb feature.

After that we used a TFidf transformation on the resulted matrix, which we will call **tf_mat1**.

For the lexical approach, we converted the resulted distance measures between the question pairs to an array, which we will call **dist_fea**.

Afer that we combined these two matrix **tf_mat1** and **dist_fea**, which we will call **tf_train**, that will be used as input to the LSVC classifier.

This combination has permitted us to have our best performance in this shared task with an average score of 83.13%. Whereas the neural approach has given use our 2nd best model with an average score of 81.50%. Table 9 present a comparison of our two best models against the benchmark.

5 Conclusion

In this paper, we presented *ST NSURL 2019 Shared Task: Semantic Question Similarity in Arabic* that participated in the *2019 NSURL Shared*

Task 8 (Semantic Question Similarity in Arabic). The performance of our best run on the test data for this Task ranked 7 between 9 teams in both private and public data sets. In this approach, we used a Linear Support Vector classifier by utilizing a combination of word, char and char_wb n-gram as features as well as a lexical approach-based model (Smith-Waterman), plus a PosTagger process.

Despite the simplicity of these features, we got promising results which encourage us to do further experiments on other features such us LSA that may lead to better results.

References

Lahsen Abouenour, Karim Bouzouba, and Paolo Rosso. 2010. An evaluated semantic query expansion and structure-based approach for enhancing arabic question/answering. *International Journal on Information and Communication Technologies*, 3(3):37–51.

Yassine Benajiba, Paolo Rosso, and Abdelouahid Lyhyaoui. 2007. Implementation of the arabiqa question answering system's components. In *Proc. Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS-2007, Fez, Morroco, April*, pages 3–5.

W Brini, M Ellouze, and L Hadrich Belguith. 2009. Qasal: Un système de question-réponse dédié pour les questions factuelles en langue arabe. *9ème Journées Scientifiques des Jeunes Chercheurs en Génie Electrique et Informatique, Tunisia*.

Davide Buscaldi, Paolo Rosso, José Manuel Gómez-Soriano, and Emilio Sanchis. 2010. Answering questions with an n-gram based passage retrieval engine. *Journal of Intelligent Information Systems*, 34(2):113–134.

Bassam Hammo, Hani Abu-Salem, and Steven Lytinen. 2002. Qarab: A question answering system to support the arabic language. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, pages 1–11. Association for Computational Linguistics.

Mohamed Lichouri, Mourad Abbas, Abed Alhakim Freihat, and Dhiya El Hak Megtouf. 2018. Word-level vs sentence-level language identification: Application to algerian and arabic dialects. *Procedia Computer Science*, 142:246–253.

FA Mohammed, Khaled Nasser, and HM Harb. 1993. A knowledge based arabic question answering system (aqas). *ACM SIGART Bulletin*, 4(4):21–30.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344.

Haitham Seelawi, Ahmad Mustafa, Al-Bataineh Hesham, Wael Farhan, and Hussein T. Al-Natsheh. 2019. NSURL-2019 task 8: Semantic question similarity in arabic. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.