Question Similarity in Community Question Answering: A Systematic Exploration of Preprocessing Methods and Models

Florian Kunneman[A], **Thiago Castro Ferreira**[B], **Emiel Krahmer**[B], and **Antal van den Bosch**[A,C]

[A]Centre for Language Studies, Radboud University, The Netherlands
[B]Tilburg center for Cognition and Communication, Tilburg University, The Netherlands
[C]KNAW Meertens Institute, Amsterdam, The Netherlands

Abstract

Community Question Answering forums are popular among Internet users, and a basic problem they encounter is trying to find out if their question has already been posed before. To address this issue, NLP researchers have developed methods to automatically detect question-similarity, which was one of the shared tasks in SemEval. The best performing systems for this task made use of Syntactic Tree Kernels or the SoftCosine metric. However, it remains unclear why these methods seem to work, whether their performance can be improved by better preprocessing methods and what kinds of errors they (and other methods) make. In this paper, we therefore systematically combine and compare these two approaches with the more traditional BM25 and translation-based models. Moreover, we analyze the impact of preprocessing steps (lowercasing, suppression of punctuation and stop words removal) and word meaning similarity based on different distributions (word translation probability, Word2Vec, fastText and ELMo) on the performance of the task. We conduct an error analysis to gain insight into the differences in performance between the system set-ups. The implementation is made publicly available.[1]

1 Introduction

Community Question Answering (CQA) forums, such as Quora[2] and Yahoo Answers[3], are popular outlets to ask questions and receive answers, as well as to browse through questions and answers. Given the large amount of material on these platforms, a basic problem users encounter is trying to find out if (a variant of) their question has already been posed (and possibly answered) before. Given a target question provided by a user, the automatic task of querying and ranking semantically similar, relevant alternative questions in CQA forums is called *question similarity*.

For efficiency reasons, the question similarity task (also known as question relevance) normally works in two ranking steps. Given a target question, the first step consists of retrieving relevant questions using a general information retrieval technique, such as BM25 (Robertson et al., 2009), or a search engine such as Google (Page et al., 1999). The second step, the focus of this work and of most other studies in this field, consists of re-ranking the most likely candidate questions with a more fine-grained, domain-specific approach. Optionally, the system could also return whether a candidate question is a duplicate of the query.

The reranking task has been included as a benchmark task (Task 3 - Subtask B) in SemEval-2016/2017 (Nakov et al., 2016, 2017). Using the domain of *Qatar Living*[4], it consisted of re-ranking ten candidate questions retrieved by Google for a target question. Several promising approaches were proposed for this challenge, most notably *SimBOW* (Charlet and Damnati, 2017) based on the SoftCosine metric and winner of SemEval-2017, and KeLP (Filice et al., 2016), which is based on Tree Kernels and provided top results for all the subtasks in the challenge. However, little is known about the effects of particular design choices for these models, especially concerning the preprocessing methods and word-similarity metrics. Moreover, we know little about

[1] https://github.com/fkunneman/DiscoSumo/tree/master/ranlp
[2] https://www.quora.com/
[3] https://answers.yahoo.com/
[4] https://www.qatarliving.com

Proceedings of Recent Advances in Natural Language Processing, pages 593–601,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_070

how these models perform in comparison to (or combined with) more traditional question similarity techniques.

In this paper we therefore systematically combine and compare the SoftCosine metric and the Syntactic Tree Kernels with the more traditional BM25 and translation-based models. Moreover, we analyze the impact of preprocessing steps (lowercasing, suppression of punctuation and stop words removal) and word-similarity metrics based on different distributions (word translation probability, Word2Vec, fastText and ELMo).

The experiments were mainly conducted on the data of SemEval 2016-2017 - Task 3, based on the Qatar Living corpus. As a secondary goal, we also evaluated our main models in classifying question duplicates on the Quora dataset, so as to assess whether the results that we find apply to different datasets.

Results show that the choice of a preprocessing method and a word-similarity metric have a considerable impact on the final results. We also show that the combination of all the analyzed approaches leads to results competitive with related work in question-similarity.

2 Models

We compare two traditional and two recent approaches in this study: BM25, Translation-Based Language Model (TRLM), SoftCosine and Smoothed Partial Tree Kernels (SPTK - Syntactic Tree Kernels).

BM25 is a fast information retrieval technique (Robertson et al., 2009) used as a search engine in the first step of the shared task by many studies. We used the implementation of BM25 provided by *gensim*[5] as a baseline.

Translation-Based Language Model (TRLM) is a question similarity ranking function, first introduced by Xue et al. (2008). The method combines a language model with a word translation system technique, and is known to obtain better results on the question similarity task than BM25 and only the language model (Jeon et al., 2005). Equation 1 summarizes the TRLM ranking score between questions Q_1 and Q_2:

$$TRLM(Q_1, Q_2) = \prod_{w \in Q_1} (1-\sigma) P_{tr}(w|Q_2) + \sigma P_{lm}(w|C)$$
$$P_{tr}(w|Q_2) = \alpha \sum_{t \in Q_2} Sim(w,t) P_{lm}(t|Q_2) + (1-\alpha) P_{lm}(w|Q_2) \quad (1)$$

$Sim(w,t)$ denotes a similarity score among words w and t. In the original study, this similarity metric is the word-translation probability $P(w|t)$ obtained by the IBM Translation Model 1 (Brown et al., 1993). Furthermore, C denotes a background corpus to compute unigram probabilities in order to avoid 0 scores.

SoftCosine is the ranking function used by *SimBOW* (Charlet and Damnati, 2017), the winning system of the question similarity re-ranking task of SemEval 2017 (Nakov et al., 2017). The method is similar to a cosine similarity between the tf-idf bag-of-words of the pair of questions, except that it also takes into account word-level similarities as a matrix M. Given X and Y as the respective tf-idf bag-of-words for questions Q_1 and Q_2, Equation 2 summarizes the SoftCosine metric.

$$SoftCos(X,Y) = \frac{X^t M Y}{\sqrt{X^t M X}\sqrt{Y^t M Y}}$$
$$X^t M X = \sum_{i=1}^{n}\sum_{j=1}^{n} X_i M_{ij} Y_j \quad (2)$$
$$M_{ij} = max(0, cosine(V_i, V_j))^2$$

As $Sim(w,t)$ in Equation 1, M_{ij} represents the similarity between the i-th word of question Q_1 and the j-th one in question Q_2. $cosine$ is the cosine similarity, and V_i and V_j are originally 300-dimension embedding representations of the words, trained on the unannotated part of the Qatar living corpus using Word2Vec (Mikolov et al., 2013) with a context window size of 10.

Smoothed Partial Tree Kernels (SPTK) are the basis of KeLP (Filice et al., 2016), a system introduced by Croce et al. (2011). SPTK applies the kernel trick by computing the similarity of question pairs based on the number of common substructures their parse trees share. The difference with Partial Tree Kernels (PTK) (Moschitti, 2006) is that SPTK also considers word relations.

Besides the different variations of the model, which are well explained in Moschitti (2006) and

[5] https://radimrehurek.com/gensim/summarization/bm25.html

Filice et al. (2016), we designed SPTK in the following form. Equation 3 portrays the notation of the similarity metric among two questions' constituency trees, i.e. T_{Q_1} and T_{Q_2}.

$$TK(T_{Q_1}, T_{Q_2}) = \sum_{n_1 \in N_{T_{Q_1}}} \sum_{n_2 \in N_{T_{Q_2}}} \Delta(n_1, n_2) \quad (3)$$

$N_{T_{Q_1}}$ and $N_{T_{Q_2}}$ are the respective sets of nodes of parse trees T_{Q_1} and T_{Q_2}. $\Delta(n_1, n_2)$ is computed in distinct forms according to three conditions. (1) If the production rules of T_{Q_1} on n_1 and T_{Q_2} on n_2 are different, then $\Delta(n_1, n_2) = 0$. (2) If n_1 and n_2 are similar preterminals, then $\Delta(n_1, n_2) = Sim(w_{n_1}, w_{n_2})$, where Sim is similar to M_{ij} in Equation 2, as well as w_{n_1} and w_{n_2} are the terminal words for n_1 and n_2, respectively. (3) If the production rules of T_{Q_1} on n_1 and T_{Q_2} on n_2 are the same and both are not preterminals, then

$$\Delta(n_1, n_2) = \prod_{j=1}^{child(n_1)} \Delta(child(n_1)_j, child(n_2)_j) \quad (4)$$

So given a pair of constituency tree questions $p = \langle T_{Q_1}, T_{Q_2} \rangle$ to have their relevance scored and a training set of pair trees C, features are extracted in the following way:

$$SPTK(T_{Q_1}, T_{Q_2}) = \{TK(T_{Q_1}, T_{c_1}) + TK(T_{Q_2}, T_{c_2})\} \quad (5)$$

where $\langle T_{c_1}, T_{c_2} \rangle_i \in C$

The extracted kernel is used in Support Vector Machines Φ, whose output decision function is the relevance score among T_{Q_1} and T_{Q_2}.

Ensemble is the method we propose to combine the relevance scores produced by the previous approaches into a single model. Given questions Q_1 and Q_2 , we trained a Logistic Regression $\phi(Q_1, Q_2)$ with the relevance scores of BM25, TRLM and SoftCosine as features:

$$Ensemble(Q_1, Q_2) = \phi(Q_1, Q_2) \quad (6)$$

After empirically testing different settings, we found that the integration of SPTK in the ensemble method was most effective when interpolating its relevance score separately with the outcome of formula 6. Equation 7 denotes the model:

$$EnsSPTK(Q_1, Q_2) = \gamma\phi(Q_1, Q_2) + (1-\gamma)\Phi(SPTK(T_{Q_1}, T_{Q_2})) \quad (7)$$

We will compare the performance of the ensemble implementation with and without SPTK. For distinction, in the following sections we will refer to the former ensemble method as *Ensemble* and the latter as *EnsSPTK*.

3 Experiments

3.1 Data

Qatar Living We ran our experiments on the data of SemEval 2016-2017 - Task 3 based on the Qatar Living corpus[6]. Its training split consists of 267 target questions, 2,669 related questions (around 10 for each target question), and 26,690 comments (around 10 per related question). The development split and test sets of 2016 and 2017 have 50, 70, and 88 target questions, respectively (with the same proportion of related questions and comments as the training set). Given a target question, each of its related questions, retrieved by Google, was manually annotated as "Perfect Match", "Relevant" or "Irrelevant". The shared-task also provided a large unannotated dataset of Qatar Living, with 189,941 questions and 1,894,456 comments. In the Qatar Living corpus, each question is formed by a subject and a body. For the models BM25, TRLM and SoftCosine, we treat a question combining the subject and body into a single text, whereas we only use the subject for SPTK.

Quora To mitigate duplicate question pages at scale, Quora motivated the development of automated ways of detecting these questions by releasing a dataset with 400,000 pairs of questions together with a label for each entry indicating whether they are semantically identical (i.e., duplicates) or not[7]. We used this dataset to evaluate our most relevant models in the task of detecting question duplicates.

3.2 Settings

For the Translation model (TRLM), C was computed based on the training questions of the dataset used in the evaluation (e.g., Qatar Living or

[6] http://alt.qcri.org/semeval2017/task3/index.php?id=data-and-tools

[7] http://qim.fs.quoracdn.net/quora_duplicate_questions.tsv

Quora). In the Qatar Living corpus, the unannotated part of the data is also used to compute C.

Across the experiments, hyperparameters of the models such as σ and α of TRLM and γ of Ensemble with SPTK were optimized in the development split of the data through Grid Search. Moreover, Support Vector Machines in SPTK and Logistic Regression in Ensemble were implemented based on the Scikit-Learn toolkit (Pedregosa et al., 2011) and had their hyperparameters tuned by cross-validation on the training set.

3.3 Evaluation

In the SemEval shared-task, the question-similarity task was treated as a binary classification task, where the models aim to predict whether a related question is "Perfect Match/Relevant" or "Irrelevant". We evaluate the models using the Mean Average Precision (MAP) as the main metric, and also report F-Score for the classification models. In the Quora dataset, we evaluated the performance of our models in predicting question duplicates also using the F-Score measure.

3.4 Experiment 1: Preprocessing

From the top models for question similarity, little is known about the design process of their preprocessing methods. Filice et al. (2016) do not report on the preprocessing that they applied, and Charlet and Damnati (2017) lowercased the text as well as removed stopwords and punctuation. So in our first experiment, we evaluated BM25, TRLM, SoftCosine, *Ensemble* and *EnsSPTK* with 3 preprocessing methods (and all combinations of them): lowercasing, removal of stopwords[8] and suppression of punctuation. For SPTK we only apply lowercasing, since its constituency trees contain punctuation and stopwords as terminals. The preprocessing methods were applied in the training, development, test and unannotated parts of the data, such that probabilities and word distributions (e.g., word translation probability, Word2Vec, etc.) were affected.

3.5 Experiment 2: Word-Similarity

A central component of all of the evaluated models except BM25 is the use of a word-similarity metric. To evaluate which distribution better captures the similarity between two words for the task, we evaluated all the models using the word-translation probabilities, plus the cosine similarity measure depicted in Equation 2. In the latter, besides Word2Vec representations, we also tested fastText (Bojanowski et al., 2017), a distribution which takes character-level information and tends to overcome spelling variations, and the top layer of ELMo (Peters et al., 2018). To equalize the trials, the data used by the models were lowercased and stripped of stop words and punctuation.

[8]We used the list of English stopwords provided by the NLTK framework (Bird and Loper, 2004)

4 Results

The first section of Table 1 lists the MAP of the preprocessing methods in the development part of the corpus for each model. Although the best combination of preprocessing methods differs between models, we see that preprocessing the data is beneficial for the performance of all models, except for SPTK. Between the best results, we see that suppression of punctuation is beneficial for all the models, while the removal of stopwords and lowercasing are detrimental to BM25 and TRLM, respectively.

The lower part of Table 1 lists the performance of each model according to the different word-level similarity metrics. The use of Word translation probabilities appears the under-performing method out of the five, showing the power of the continuous word representations. Surprisingly, we do not observe an improvement of fastText over Word2Vec representations. Even though CQA forums may have very noisy texts, the character-level information that fastText takes into account apparently does not help. Using the top layer of ELMo concatenated with Word2Vec representations leads to the best results in encoding the relation between words, except with TRLM.

Final Results Table 2 lists the results of the models with their best settings in the test sets of SemEval 2016-2017: BM25 with lowercased data without punctuation; TRLM with Word2Vec without stop words and punctuation; SoftCosine with Word2Vec+ELMo, lowercased data without stop words and punctuation; and SPTK with Word2Vec+ELMo and lowercased data. The table also shows the results of the best baseline (e.g., Google) and the winners of the SemEval 2016-2017 challenges. As expected, our best models were the ensemble approaches (e.g., *Ensemble* and *EnsSPTK*), which combine the ranking scores of all the other evaluated approaches and outperform

Preproc.	BM25	TRLM	SoftCosine	SPTK	Ensemble	EnsSPTK
L.S.P.	68.80	68.43	**72.75**	-	**71.62**	**72.40**
L.S.	67.31	63.25	69.15	-	69.50	71.29
L.P.	**69.95**	68.42	65.33	-	68.70	69.16
S.P.	66.03	**68.65**	68.56	-	68.67	70.37
L.	67.07	66.42	63.68	54.34	67.04	67.41
S.	63.77	64.53	67.01	-	67.85	68.36
P.	65.05	64.38	60.04	-	65.31	66.66
-	63.52	64.95	60.66	**54.44**	63.08	64.31
Metric	**BM25**	**TRLM**	**SoftCosine**	**SPTK**	**Ensemble**	**EnsSPTK**
Translation	-	68.43	70.75	48.10	70.80	70.80
Word2Vec	-	**72.90**	72.75	54.44	71.40	72.64
fastText	-	70.93	71.07	53.49	71.92	71.92
Word2Vec+ELMo	-	71.41	**73.89**	**54.78**	**73.90**	**74.63**
fastText+ELMo	-	70.56	73.43	54.77	73.73	73.73

Table 1: MAP results on the different preprocessing and word-relation metric conditions in the development set. In the first part, *L.*, *S.* and *P.* denote lowercase, stop words removal and punctuation suppression methods respectively.

	2016		2017	
Models	**MAP**	**F-1**	**MAP**	**F-1**
Baseline	74.75	-	41.85	-
BM25	73.33	-	44.98	-
TRLM	71.94	-	44.25	-
SoftCosine	74.10	-	45.23	-
SPTK	45.61	21.24^{C}	29.63	33.13^{C}
Ensemble	75.48	66.96^{B}	46.74	$\mathbf{48.74}^{A}$
EnsSPTK	75.40	$\mathbf{68.34}^{A}$	47.06	$\mathbf{48.72}^{A}$
Winner	**76.70**	66.39^{B}	**47.22**	42.37^{B}

Table 2: Final results of the models with their best preprocessing and word-relation settings in the test sets of SemEval 2016-2017. F-Score results were statistically significant with $p < 0.05$ according to the McNemar's test, with A outperforming B and C, and B outperforming C.

	No preproc. (-)	Preproc. (L.S.P.)
Word2Vec	0.50	0.50
Word2Vec+ELMo	0.50	**0.52***

Table 3: F1 scores of our ensemble method with different preprocessing techniques and word similarity measures on the Quora dev set.

the competitive baselines of SemEval 2016 and 2017.

Regarding the comparison between *Ensemble* and *EnsSPTK* in the test set of Semeval 2016, the approach without SPTK (*Ensemble*) is slightly better on re-ranking similar questions, but is significantly worse on classifying duplicates than *EnsSPTK*. The results are different in the test set of Semeval 2017: the latter approach is slightly better than the former on re-ranking similar questions according to the MAP metric, but shows a non-significant difference in classifying duplicates according to the F-1 score metric.

Although the results between our two best approaches are inconclusive, we argue that the inclusion of SPTK in the ensemble is not beneficial due to the trade-off between efficiency and performance. The SPTK approach, mainly its kernel, is computationally expensive and does not considerably improve the performance of the ensemble. For efficiency reasons we elect *Ensemble* as our best approach.

Checking the coefficients of the trained logistic regression model of *Ensemble*, we saw that the BM25 score (with a coefficient of 4.13) is the most relevant feature of the model, shortly followed by the SoftCosine score (with a coefficient of 3.48) and finally by the TRLM one (with a coefficient of 1.1).

In SemEval-2016, the UH-PRHLT model was the winner of the shared-task (Franco-Salvador et al., 2016). This system is based on a range of lexical (cosine similarity, word, noun and n-gram overlap) and semantic (word representations, alignments, knowledge graphs and common frames) features. In turn, our best model, *Ensemble*, with considerably less features, obtains competitive results in terms of MAP. The same pattern is seen for the SemEval-2017 test set: the Ensemble approach obtained competitive results with the winner *SimBOW*, also based on the SoftCosine metric, in terms of MAP, and outperforms it in F-Score.

Quora results Based on the previous results, we also evaluated the performance of our best question-similarity model, *Ensemble*[9], in classifying question *duplicates* on the Quora dataset. Table 3 depicts the results of our ensemble method with and without preprocessing and using two similarity metrics (Word2Vec and Word2Vec+ELMo). The best F-Score was obtained by the version which preprocesses the questions and represents the words with Word2Vec+ELMo. Results were statistically significant with $p < 0.05$ according to the McNemar's test.

5 Error Analysis

To obtain insight into the improvement by preprocessing setting, in Figure 1 we present the percentage of similar questions that were ranked better (placed on a higher position formerly occupied by a non-similar), equally or worse (switched a lower position with a non-similar) for each model-preprocessing combination in the Qatar Living corpus. The graph shows that each preprocessing manipulation results in both improved rankings and worsened rankings. The model that is least affected by the preprocessing steps is BM25, which shows to be a stable baseline. Most gain is seen for the SoftCosine model with all preprocessing steps, where 38% of the duplicates are ranked better and only 9% is ranked lower than a non-duplicate. Regarding the preprocessing steps applied in isolation, lowercasing leads to most changes for TRLM and BM25, while SoftCosine is most affected after removing stopwords.

The changes in performance by similarity metric are also presented in Figure 1. The highest gains are seen for the TRLM model, which yields an improved ranking for over 20% of the duplicates and a poor re-ranking for 12% to 14% when a similarity metric other than alignment is used. The SPTK model is not helped by a different similarity metric, with the most detrimental effect when combining the model with the translation alignment or fastText. The SoftCosine model with the default Word2Vec is also rather robust, with only a slight improvement when applying one of the ELMo metrics. These metrics do affect the rankings considerably, but lead to fairly equal improvements and declines of the ranking quality.

[9]Given the size of the Quora dataset, computing the kernel trick of SPTK would be intractable.

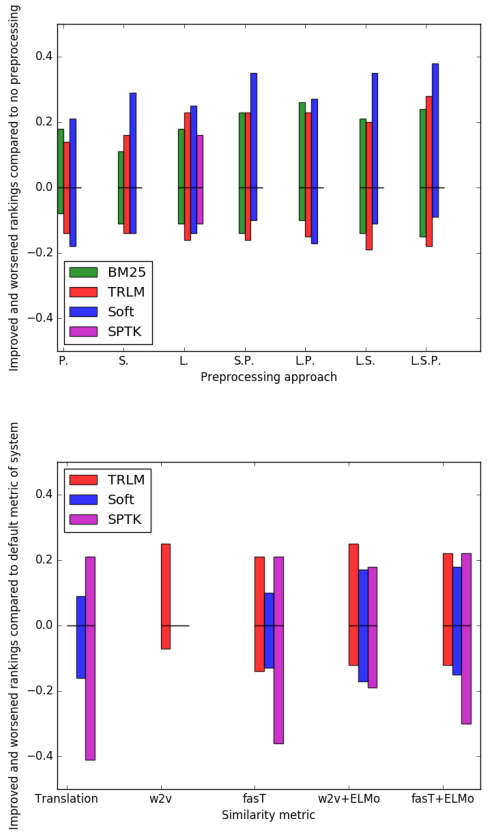

Figure 1: Percentage of similar questions that were ranked better, equally or worse after any of the preprocessing manipulations or similarity metrics combined with each system, in comparison to the standard setting without preprocessing (first graph) or the standard similarity metric for each system (second graph).

The performance patterns presented in Figure 1 show that the SoftCosine metric is affected most by the presence of stopwords. Explicit evidence is presented in Figure 2, which depicts the scores of the SoftCosine settings with and without preprocessing in relation to the number of stopwords in a question-pair. The setting without preprocessing shows a correlation with the number of stopwords: the similarity score goes up as the number of stopwords increases. The setting with preprocessing is, as expected, robust to the number of stopwords. This shows that the SoftCosine metric is considerably affected by the inclusion of stopwords, which hampers performance for the task of question similarity.

In Table 4 we present examples of question pairs in the Qatar Living development set along with their Gold standard label and the preprocessing steps or model that yielded a proper ranking

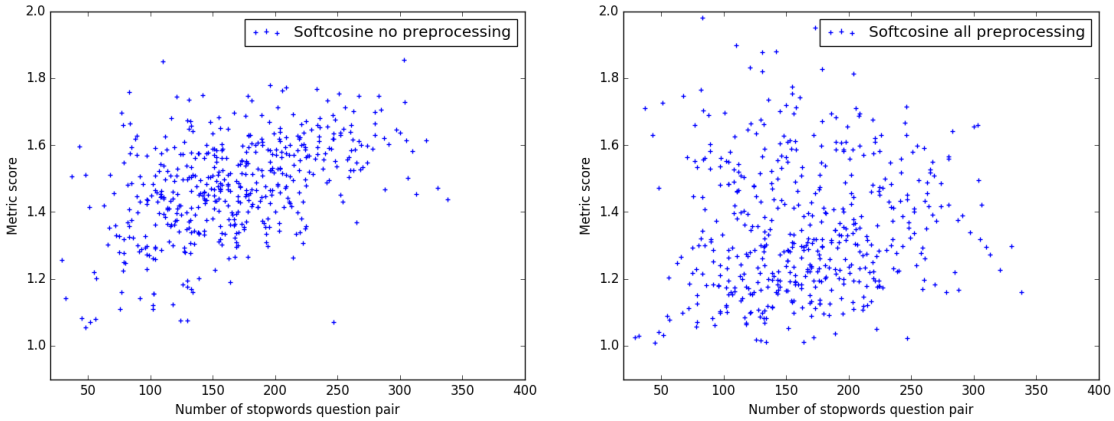

Figure 2: Similarity score in relation to the number of stopwords in a question pair, for the SoftCosine settings with and without preprocessing.

Question ID	Target question (subject - body)	Related question (subject - body)	Gold standard	Best performance
Q314	QLing after working hours - how many of you logon to QL after your working hours?	Surfing QL during office hours - How many hour(s) everyone spend surfing QL during office hours?	Similar	Stopword removal; SoftCosine
Q278	Beach cleaning - I am planning to organize a community service specifically beach cleaning to be carried out by our company staff. Any good suggestion of a beach?	NOT ONE PUBLIC BEACH IN DOHA? - Surrounded by pleasant waters and not ONE public beach! Ridiculous.	Not similar	No lowercasing
Q293	Water theme park in qatar - Hai Friends..... Any one knows the location of watar theme park in qatar; Is it beatiful? childrens have enough ride?? and howmuch fee Thanks	any water theme park in qatar? - Do you know about any water theme park in qatar?	Similar	SPTK

Table 4: Examples of question pairs with particular performance patterns.

for this pair. The first example, with question ID Q314, is of a similar question pair that was most often ranked in a high position by settings that included stopword removal and the SoftCosine model. Stopword removal shows particularly effective for the target question by removing 7 of the 15 words, and SoftCosine best matches 'office hours' to 'working hours'. The second question pair is exemplary of cases where preprocessing is actually detrimental. The related question, not similar to the target question, is partly written in capitals. After lowercasing, the word 'beach' is matched with the target question, which might result in a higher similarity score than questions that are actually similar. The final example, with ID Q293, is particularly well ranked by the SPTK model. On its own, SPTK did not compete with the other models in our study, but the focus on syntactic tree kernels could add a valuable angle to the similarity assessment. In this example, the good assessment by SPTK is likely due to the central phrase 'watar theme park in qatar', which is recurring, albeit with a different spelling, in the related question.

6 Discussion and Conclusion

Until now, careful preprocessing and smart combining of methods have remained understudied in the field of community question answering. Our results highlight that both pay off, yielding state-of-the-art results. Our findings show that lowercasing the input and removing both punctuation and stopwords yields the most robust outcomes, especially for the SoftCosine metric. In addition, representing the meaning of words by means of Word2Vec combined with the top layer of ELMo is the most beneficial word similarity implementation. Combining several metrics implemented with these optimal settings into an ensemble system based on logistic regression yields the best

performance in terms of F1-score, being competitive with the winners of the SemEval tasks, and using fewer components.

The error analysis showed that the BM25 model is most stable across different preprocessing metrics, while the SoftCosine model mostly profits from preprocessing. Given the semantic matching that is done as part of SoftCosine and is absent in BM25, we can infer that preprocessing is an important prerequisite for effectively ranking question pairs based on semantic links.

Most of our experimentation was conducted on the Semeval dataset, in which similarity between questions is labeled. We also showed that adjusting preprocessing and word similarity settings leaded to better results in the task of identifying question duplicates, in the Quora dataset. More research is needed to see whether the patterns that we find are dataset-independent.

In future work we aim to compare the optimal models from our current study in a real-world setting, by running A/B testing on a open-domain CQA platform. Through clicks and likes by the users of such a platform, we can obtain insights into the value of these models when applied in the wild with many different question topics.

Acknowledgements

This work is part of the research programme Discussion Thread Summarization for Mobile Devices, which is financed by the Netherlands Organisation for Scientific Research (NWO). We thank the reviewers for their valuable comments.

References

Steven Bird and Edward Loper. 2004. Nltk: The natural language toolkit. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACLdemo '04. https://doi.org/10.3115/1219044.1219075.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146. http://aclweb.org/anthology/Q17-1010.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Meredith J Goldsmith, Jan Hajic, Robert L Mercer, and Surya Mohanty. 1993. But dictionaries are data too. In *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, pages 202–205.

Delphine Charlet and Geraldine Damnati. 2017. Simbow at semeval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pages 315–319.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1034–1046.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pages 1116–1123.

Marc Franco-Salvador, Sudipta Kar, Thamar Solorio, and Paolo Rosso. 2016. 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)*. Association for Computational Linguistics, pages 814–821. https://doi.org/10.18653/v1/S16-1126.

Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, pages 84–90.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 746–751.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *European Conference on Machine Learning*. Springer, pages 318–329.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 27–48. http://www.aclweb.org/anthology/S17-2003.

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-2016)*. Association for Computational

Linguistics, San Diego, California, pages 525–545. http://www.aclweb.org/anthology/S16-1083.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, pages 2227–2237. https://doi.org/10.18653/v1/N18-1202.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval* 3(4):333–389.

Xiaobing Xue, Jiwoon Jeon, and W Bruce Croft. 2008. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 475–482.