SemanticZ at SemEval-2016 Task 3: Ranking Relevant Answers in Community Question Answering Using Semantic Similarity Based on Fine-tuned Word Embeddings

Todor Mihaylov
Research Training Group AIPHES
Institute for Computational Linguistics
Heidelberg University
mihaylov@cl.uni-heidelberg.de

Preslav Nakov
Qatar Computing Research Institute, HBKU
P.O. box 5825
Doha, Qatar
pnakov@qf.org.qa

Abstract

We describe our system for finding good answers in a community forum, as defined in SemEval-2016, Task 3 on Community Question Answering. Our approach relies on several semantic similarity features based on fine-tuned word embeddings and topics similarities. In the main Subtask C, our primary submission was ranked third, with a MAP of 51.68 and accuracy of 69.94. In Subtask A, our primary submission was also third, with MAP of 77.58 and accuracy of 73.39.

1 Introduction

Posting questions that have already been asked and answered in a community forum is annoying to users as it usually ends up with them being referred to a previously asked question. The SemEval-2016 Task 3 on Community Question Answering[1] (Nakov et al., 2016) aims to solve this real-life problem. The main subtask (Subtask C) asks to find an answer that already exists in the forum and will be appropriate as a response to a newly-posted question. There is also a secondary, Subtask A, which focuses on Question-Comment Similarity and asks to rank the comments within a question-comment thread based on their relevance with respect to the thread's question.

Here, we examine the performance of using different word embeddings obtained with the Word2Vec tool (Mikolov et al., 2013), which we use to build vectors for the questions and the answers. We train classifiers using features derived from these embeddings to solve subtasks A and C.

Our contribution is in producing good word embeddings based on empirical evaluation of different configurations working in the Community Question Answering domain; as they perform well, we make them freely available to the research community.[2]

2 Related Work

This year's SemEval-2016 Task 3 is a follow up of SemEval-2015 Task 3 on Answer Selection in Community Question Answering (Nakov et al., 2015). The 2015 subtask A asked to determine whether an answer was relevant, potentially useful, or bad, while this year this is about ranking.

Here we focus on features that use semantic knowledge such as word embeddings, various features extracted from word embeddings, and topic models. Word embeddings and word embeddings similarities have been used by teams in the 2015 edition of the task (Belinkov et al., 2015; Zamanov et al., 2015; Tran et al., 2015; Nicosia et al., 2015). LDA topic have also been used (Tran et al., 2015).

Many other features have been tried for the task. For example, Tran et al. (2015) used metadata about the question and the comment. User profile statistics such as number of *Good*, *Bad* and *Potentially Useful* comments by a given user have been used to model user likelihood of posting different types of comment (Nicosia et al., 2015). Vo et al. (2015) and Nicosia et al. (2015) used syntactic tree similarities to compare questions to comments. The problem of selecting relevant answers has even been approached as a spam filtering task (Vo et al., 2015).

[1] http://alt.qcri.org/semeval2016/task3/

[2] https://github.com/tbmihailov/semeval2016-task3-cqa

3 Data

In our experiments, we used annotated training, development and testing datasets, as well as a large unannotated dataset, all provided by the SemEval-2016 Task 3 organizers. We further collected some additional unannotated in-domain data from some other sources, as explained below; finally, we used some models pretrained on out-of-domain data.

Training, development, and testing data. For Subtask A, there are 6,398 questions and 40,288 comments from their question-answer threads, and for Subtask C, there are 317 original questions, 3,169 related questions, and 31,690 comments. For both subtasks, the comments are annotated as *Good*, *PotentiallyUseful* and *Bad*; for subtask A, the annotation is with respect to the question in whose thread the comment appeared, while for subtask C, it is with respect to a new question. For both subtasks, a successful ranking is one that ranks all *Good* comments before all *PotentiallyUseful* and *Bad* ones (without distinguishing between the latter two).

Unannotated data. We performed experiments with Word2Vec embeddings trained on different unannotated data sources. We wanted to find the best performing embeddings and to use them in our system. In Table 1, we list the various data sources we used for training our Word2Vec models, and their vocabulary size.

Qatar Living Forum is the original Qatar Living.[3] unannotated data containing 189,941 questions and 1,894,456 comments. It is limited to the forums section of the Qatar Living website.

Qatar Living Forum + Ext includes the *Qatar Living Forum* dataset, i.e., the forums, but also some other sections of Qatar Living: Jobs, Classifieds, Pages, Wiki and Events posts.

Doha News is a dataset that we built by crawling about 7,000 news publications about the life in Doha, Qatar from the DohaNews website.[4]

We also used an out-of-domain, general model, which is readily-pretrained using Word2Vec on ***Google News***,[5] as provided by Mikolov et al. (2013).

Features	Train size	Vocab
Qatar Living Forum	61.84M	104K
Qatar Living Forum+Ext	90M	126K
Google News	100B	3M
Doha News	1.45M	17K

Table 1: Data used for training word embedding vectors. Shown are training source size (word tokens) and vocabulary size (word types).

4 Method

Below we focus our explanation on subtask A; for subtask C, we combine the predictions for subtask A with the Google's reciprocal rank for the related question (see below).

We approach subtask A as a classification problem. For each comment, we extract variety of features from both the question and the comment, and we train a classifier to label comments as Good or Bad with respect to the thread question. We rank the comments in each question according to the classifier's score of being classified as Good with respect to the question.

We first train several word embedding vector models and we fine-tune them using different configurations. For fine-tuning the parameters of the word embeddings training configuration, we setup a simple baseline system and we evaluate it on the official MAP score. We then use the best-performing embeddings in our further experiments. Our main features are semantic similarity based on word embeddings and topics, but we also use some metadata features.

4.1 Preprocessing

Before extracting features, we preprocessed the input text using several steps. We first replaced URLs in text with TOKEN_URL, numbers with TOKEN_NUM, images with TOKEN_IMG, and emoticons with TOKEN_EMO. We then tokenized the text by matching only continuous alphabet characters including _(underscore). Next, we lowercased the result. For the training, the development, and the test datasets, we removed the stopwords using the English stopwords lexicon from the NLTK toolkit (Bird and Loper, 2004).

[3] www.qatarliving.com is an online community for everyone living in or interested in the State of Qatar.

[4] dohanews.co covers breaking news, politics, business, culture and more in and around Qatar.

[5] code.google.com/archive/p/word2vec/

4.2 Features

We used several semantic vector similarity and metadata feature groups. For the similarity measures mentioned below, we used cosine similarity:

$$1 - \frac{u.v}{\|u\| . \|v\|} \qquad (1)$$

Semantic Word Embeddings. We used semantic word embeddings obtained from Word2Vec models trained on different unannotated data sources including the QatarLiving and DohaNews. We also used a model pre-trained on Google News text. For each piece of text such as comment text, question body and question subject, we constructed the centroid vector from the vectors of all words in that text (excluding stopwords).

$$centroid(w_{1..n}) = \frac{\sum_{i=1}^{n} w_i}{n} \qquad (2)$$

We built centroid vectors (2) from the question body and the comment text. We then examined different Word2Vec models in terms of training source and training configuration including word vector size, training window size, minimum word occurrence in the corpus, and number of skip-grams.

Semantic Vector Similarities. We used various similarity features calculated using the centroid word vectors on the question body, on the question subject and on the comment text, as well as on parts thereof:

Question to Answer similarity. We assume that a relevant answer should have a centroid vector that is close to that for the question. We used the question body to comment text, and question subject to comment text vector similarities.

Maximized similarity. We ranked each word in the answer text to the question body centroid vector according to their similarity and we took the average similarity of the top N words. We took the top 1,2,3 and 5 words similarities as features. The assumption here is that if the average similarity for the top N most similar words is high, then the answer might be relevant.

Aligned similarity. For each word in the question body, we chose the most similar word from the comment text and we took the average of all best word pair similarities as suggested in (Tran et al., 2015).

Part of speech (POS) based word vector similarities. We performed part of speech tagging using the Stanford tagger (Toutanova et al., 2003), and we took similarities between centroid vectors of words with a specific tag from the comment text and the centroid vector of the words with a specific tag from the question body text. The assumption is that some parts of speech between the question and the comment might be closer than other parts of speech.

Word clusters (WC) similarity. We clustered the word vectors from the Word2Vec vocabulary in 1,000 clusters (with 200 words per cluster on average) using K-Means clustering. We then calculated the cluster similarity between the question body word clusters and the answer text word clusters. For all experiments, we used clusters obtained from the Word2Vec model trained on QatarLiving forums with vector size of 100, window size 10, minimum words frequency of 5, and skip-gram 1.

LDA topic similarity. We performed topic clustering using Latent Dirichlet Allocation (LDA) as implemented in the *gensim* toolkit (Řehůřek and Sojka, 2010) on Train1+Train2+Dev questions and comments. We built topic models with 100 topics. For each word in the question body and for the comment text, we built a bag-of-topics with corresponding distribution, and calculated similarity. The assumption here is that if the question and the comment share similar topics, they are more likely to be relevant to each other.

Metadata. In addition to the semantic features described above, we also used some common sense metadata features:

Answer contains a question mark. If the comment has an question mark, it may be another question, which might indicate a bad answer.

Answer length. The assumption here is that longer answers could bring more useful detail.

Question length. If the question is longer, it may be more clear, which may help users give a more relevant answer.

Question to comment length. If the question is long and the answer is short, it may be less relevant.

The answer's author is the same as the corresponding question's author. If the answer is posted by the same user who posted the question and it is relevant, why has he/she asked the question in the first place?

Answer rank in the thread. Earlier answers could be posted by users who visit the forum more often, and they may have read more similar questions and answers. Moreover, discussion in the forum tends to diverge from the question over time.

Question category. We took the category of the question as a sparse binary feature vector (a feature with a value of 1 appears if question is in the category). The assumption here is that the question-comment relevance might depend on the category of the question.

4.3 Classifier

For each Question+Comment pair, we extracted the features explained above from the *Question* body and the subject text fields, and from the *Comment* text; we also extracted the relevant metadata. We concatenated the extracted features in a bag of features vector, scaling them in the 0 to 1 range, and feeding them to a classifier. In our experiments, we used different feature configurations. We used L2-regularized logistic regression classifier as implemented in Liblinear (Fan et al., 2008). For most of our experiments, we tuned the classifier with different values of the C (cost) parameter, and we took the one that yielded the best accuracy on 5-fold cross-validation on the training set. We used binary classification *Good* vs. *Bad* (including both *Bad* and *Potentially Useful* original labels). The output of the evaluation for each test example was a label, either *Good* or *Bad*, and the probability of being *Good* in the 0 to 1 range. We then used this output probability as a relevance rank for each *Comment* in the *Question* thread.

5 Experiments and Evaluation

As explained above, we rely mainly on semantic features extracted from Word2Vec word embeddings. Thus, we ran several experiments looking for the best embeddings for the task.

Table 2 shows experiments with Word2Vec models trained on the unannotated datasets described above. The Google News Word2Vec model comes pretrained with vector size of 300, window 10, minimum word frequency of 10 and skip-gram 1. We started with training our three Word2Vec models using the same parameters.

	Dev2016	
Dataset	**MAP**	**Accuracy**
Qatar Living Forum	**0.6311**	0.7078
Qatar Living Forum+Ext	0.6269	**0.7131**
Google News	0.6113	0.6996
Doha News	0.5769	0.6844

Table 2: Semantic vectors trained on different unannotated datasets as the only features for subtask A: training on train2016-part1, testing on dev2016.

	Test2016	
Vector size	**MAP**	**Accuracy**
800	**78.45**	74.22
700	78.12	73.98
600	77.31	73.15
500	77.61	73.30
400	78.36	74.19
300	77.25	74.50
200	77.90	73.88
100	77.08	**74.53**
50	77.22	73.85
20	75.44	72.42
Baseline	59.53	-

Table 3: Semantic vectors of different vector sizes, trained on Qatar Living Forum as features for subtask A (together with all other features): training on train2016-part1, testing on test2016.

Table 2 shows results using raw word vectors as features, together with an extra feature for question body to comment cosine similarity. We can see that training on *Qatar Living Forum* data performs best followed by using *Qatar Living Forum+Ext*, *Google News*, and *Doha News*. This is not surprising as the first two datasets are in-domain, while the latter two cover more topics (as they are news) and more formal language. Overall, Doha News contains topics that largely overlap with the topics discussed in the Qatar Living forum; yet, it uses more formal language and contains very little conversational word types (mostly in quotations and interviews); moreover, being smaller in size, it covers much less vocabulary. Based on these preliminary experiments on Dev2016, we concluded that the domain-specific word vectors trained on *Qatar Living Forum* were the best for this task, and we used them further in our experiments.

After we have selected the best dataset for training our semantic vectors, we continued with various experiments to select the best training parameters for Word2Vec. Below we present the results of these experiments on Test2016, but we experimented with Dev2016 when developing our system.

In Table 3, we present experiments with different vector sizes. We trained our classifier with all features mentioned above, extracted for the corresponding word vector model. We can see that word vectors of size 800 perform best followed by sizes 400 and 700. However, we should note that using word vectors of size 800 generates more than 1,650 features (800+800+other features), which slows down training and evaluation. Moreover, in our experiments, we noticed that using large word vectors blurs the impact of the other, non-vector features.

Thus, next we tried to achieve the MAP for the 800-size vector by using better parameters for smaller vector sizes. Table 4 shows the results, where we used vectors of size 100 and 200. We can see that the configuration with word vector size 200, window size 5, minimum word frequency 1 and skip-gram 3 performed best improving the 200 vectors MAP by 0.31 (compared to Table 3). However, the experiments with word vector size 100 improved its MAP score by 0.85, which suggests that there might be potential for improvement when using vectors of smaller size. We also tried to use Doc2Vec (Le and Mikolov, 2014) instead of Word2Vec, but this led to noticeably lower performance.

We further experimented with Word2Vec models trained with different configurations and different feature groups. Tables 5 and 6 show the results for ablation experiments using the best-performing configuration for Subtask A and C, respectively.

For Subtask A we achieved the best score with semantic vectors of size 200, trained with window size 5, minimum word frequency 1 and skip-grams 3. The best score we achived (MAP 78.52) is slightly hbetter than the best score from Table 3 (MAP 78.45), which means that it may be a good idea to use smaller word vectors in combination with other features. We can see that the features that contribute most (the bottom features are better) are the raw word centroid vectors and metadata features, followed by various similarities such as LDA topic similarity and POS-tagged-word similarity.

				Test2016	
Size	**Window**	**Freq**	**Skip**	**MAP**	**Acc**
200	5	1	3	**78.21**	74.25
200	5	5	1	78.19	73.49
200	5	5	3	78.13	74.01
200	5	1	1	78.01	**74.53**
100	5	1	1	77.93	74.19
200	10	5	1	77.90	73.88
100	5	1	3	77.81	73.94
100	10	1	1	77.72	74.43
200	10	1	1	77.58	74.25
100	5	5	1	77.53	74.07
200	10	1	3	77.43	73.73
100	10	10	1	77.18	73.79
100	10	5	1	77.08	**74.53**

Table 4: Exploring Word2Vec training parameters on Qatar Living Forum: word vector size (Size), context window (Window), minimum word frequency (Freq), and skip-grams (Skip). Vectors used as features for subtask A (together with all other features): training on train2016-part1, testing on test2016.

Train2016-part1 as training	**Test2016**	
Features	**MAP**	**Acc**
All − Quest. to Comment sim	**78.52**	74.31
All − Maximized similarity	78.38	74.59
All − Word Clusters similarity	78.29	74.25
All − WC sim & Meta cat	78.22	74.04
All − Meta categories	78.21	74.25
All	78.21	74.25
All − Meta cat & LDA sim	78.18	73.88
All − Ext POS sim & WC sim	78.10	74.28
All − Aligned similarity	77.97	74.16
All − Cat & WC & LDA sim	77.95	74.19
All − WC & LDA sims	77.92	74.25
All − Ext POS sim	77.92	74.43
All − LDA sim	77.85	74.37
All − POS sim	77.77	**74.80**
All − Metadata full	74.50	70.31
All − Word Vectors	74.35	70.80
Primary	77.58	73.39
Contrastive 1	77.16	73.88
Contrastive 2	75.41	72.26
Baseline (IR)	59.53	–

Table 5: Subtask A. Using all features without some feature groups. Word2Vec is trained with word vector size 200, context window 5, minimum word frequency 1, and skip-grams 3.

Train2016-part1 as training	**Test-2016**	
Features	**MAP**	**Acc**
All − Q to C sim	**53.39**	69.87
All − Meta categories	53.06	69.81
All − WC sim & Meta cat	52.91	69.54
All − WC sim & LDA sim	52.84	70.06
All − Meta cat & LDA sim	52.83	69.87
All − Ext POS sim & WC sim	52.82	70.21
All	52.78	69.43
All − Aligned similarity	52.76	70.10
All − Word Clusters similarity	52.58	69.63
All − Maximized similarity	52.47	69.27
All − Cat & WC & LDA sim	52.44	69.51
All − Ext POS sim	52.23	69.91
All − LDA sim	52.08	69.97
All − POS sim	51.57	69.96
All − Word Vectors	49.57	70.13
All − Metadata full	46.03	**71.06**
Primary	51.68	69.94
Contrastive 1	51.46	69.69
Contrastive 2	48.76	69.71
Baseline (IR)	28.88	–

Table 6: Subtask C. Using all features without some feature groups. Word2Vec is trained with word vector size 100, context window 5, minimum word frequency 1, and skip-grams 1.

For Subtask C, we achieved the best score with vectors of size 100, trained with window size 5, minimum word frequency 1, and skip-grams 1. The features that contributed most were mostly the same as for Subtask A. One difference is the maximized similarity features group, which now yields worse results when excluded, which indicates its importance.

Our *Primary*, *Contrastive 1* and *Contrastive 2* submissions were built with the same feature set: *All features - POS similarity & Meta Category*, but were trained with fixed C=0.55 on different datasets: *Primary* was trained on Train2016-part1, *Contrastive 1* was trained on Train2016-part1 + Train2016-part2, and *Contrastive 2* was trained on Train2016-part2.

6 Conclusion and Future Work

We have described our system for SemEval-2016, Task 3 on Community Question Answering. Our approach relied on several semantic similarity features based on fine-tuned word embeddings and topics similarities.

In the main Subtask C, our primary submission was ranked third, with a MAP of 51.68 and accuracy of 69.94. In Subtask A, our primary submission was also third, with MAP of 77.58 and accuracy of 73.39. After the submission deadline, we improved our MAP score to 78.52 for Subtask A, and to 53.39 for Subtask C, which would rank our system second.

In future work, we plan to use our best performing word embeddings models and features in a deep learning architecture, e.g., as in the MTE-NN system (Guzmán et al., 2016a; Guzmán et al., 2016b), which borrowed an entire neural network framework and achitecture from previous work on machine translation evaluation (Guzmán et al., 2015). We also want to incorporate several rich knowledge sources, e.g., as in the SUper Team system (Mihaylova et al., 2016), including troll user features as inspired by (Mihaylov et al., 2015a; Mihaylov et al., 2015b; Mihaylov and Nakov, 2016), and PMI-based goodness polarity lexicons as in the PMI-cool system (Balchev et al., 2016), as well as sentiment polarity features (Nicosia et al., 2015).

We further plan to use information from entire threads to make better predictions, as using thread-level information for answer classification has already been shown useful for SemEval-2015 Task 3, subtask A, e.g., by using features modeling the thread structure and dialogue (Nicosia et al., 2015; Barrón-Cedeño et al., 2015), or by applying thread-level inference using the predictions of local classifiers (Joty et al., 2015; Joty et al., 2016). How to use such models efficiently in the ranking setup of 2016 is an interesting research question.

Finally, we would like to address subtask C in a more solid way, making good use of the data, the gold annotations, the features, the models, and the predictions for subtasks A and B.

Acknowledgments. This work is partly supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1. It is also part of the Interactive sYstems for Answer Search (Iyas) project, which is developed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), Hamab bin Khalifa University (HBKU), part of Qatar Foundation in collaboration with MIT-CSAIL.

References

Daniel Balchev, Yasen Kiprov, Ivan Koychev, and Preslav Nakov. 2016. PMI-cool at SemEval-2016 Task 3: Experiments with PMI and goodness polarity lexicons for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, pages 687–693, Beijing, China.

Yonatan Belinkov, Mitra Mohtarami, Scott Cyphers, and James Glass. 2015. VectorSLU: A continuous word vector approach to answer selection in community question answering systems. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 282–287, Denver, Colorado, USA.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 214–217, Barcelona, Spain.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2015. Pairwise neural machine translation evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL-IJCNLP '15, pages 805–814, Beijing, China.

Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2016a. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, Berlin, Germany.

Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2016b. MTE-NN at SemEval-2016 Task 3: Can machine translation evaluation help community question answering? In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

Shafiq Joty, Alberto Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2015. Global thread-level inference for comment classification in community question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 573–578, Lisbon, Portugal.

Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2016. Joint learning with global inference for comment classification in community question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, San Diego, California, USA.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Todor Mihaylov and Preslav Nakov. 2016. Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, Berlin, Germany.

Todor Mihaylov, Georgi Georgiev, and Preslav Nakov. 2015a. Finding opinion manipulation trolls in news community forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, CoNLL '15, pages 310–314, Beijing, China.

Todor Mihaylov, Ivan Koychev, Georgi Georgiev, and Preslav Nakov. 2015b. Exposing paid opinion manipulation trolls. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, RANLP '15, pages 443–450, Hissar, Bulgaria.

Tsvetomila Mihaylova, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. SUper Team at SemEval-2016 Task 3: Building a feature-rich system for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

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*, NAACL-HLT '13, pages 746–751, Atlanta, Georgia, USA.

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. SemEval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 269–281, Denver, Colorado, USA.

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, USA.

Massimo Nicosia, Simone Filice, Alberto Barrón-Cedeño, Iman Saleh, Hamdy Mubarak, Wei Gao, Preslav Nakov, Giovanni Da San Martino, Alessandro Moschitti, Kareem Darwish, Lluís Màrquez, Shafiq Joty, and Walid Magdy. 2015. QCRI: Answer selection for community question answering - experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 203–209, Denver, Colorado, USA.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL-HLT '03, pages 173–180, Edmonton, Canada.

Quan Hung Tran, Vu Tran, Tu Vu, Minh Nguyen, and Son Bao Pham. 2015. JAIST: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 215–219, Denver, Colorado, USA.

Ngoc Phuoc An Vo, Simone Magnolini, and Octavian Popescu. 2015. FBK-HLT: An application of semantic textual similarity for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 231–235, Denver, Colorado, USA.

Ivan Zamanov, Marina Kraeva, Nelly Hateva, Ivana Yovcheva, Ivelina Nikolova, and Galia Angelova. 2015. Voltron: A hybrid system for answer validation based on lexical and distance features. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 242–246, Denver, Colorado, USA.