UINSUSKA-TiTech at SemEval-2017 Task 3: Exploiting Word Importance Levels for Similarity Features for CQA

Surya Agustian[1,2] and **Hiroya Takamura**[2]
[1]Teknik Informatika, UIN Sultan Syarif Kasim Riau, Indonesia
[1]Computational Intelligence & System Science, Tokyo Institute of Technology, Japan
[2]Institute of Innovative Research, Tokyo Institute of Technology, Japan
sagustian{@uin-suska.ac.id, @lr.pi.titech.ac.jp}, takamura@pi.titech.ac.jp

Abstract

The majority of core techniques to solve many problems in Community Question Answering (CQA) task rely on similarity computation. This work focuses on similarity between two sentences (or questions in subtask B) based on word embeddings. We exploit words importance levels in sentences or questions for similarity features, for classification and ranking with machine learning. Using only 2 types of similarity metric, our proposed method has shown comparable results with other complex systems. This method on subtask B 2017 dataset is ranked on position 7 out of 13 participants. Evaluation on 2016 dataset is on position 8 of 12, outperforms some complex systems. Further, this finding is explorable and potential to be used as baseline and extensible for many tasks in CQA and other textual similarity based system.

1 Introduction

Community Question Answering (CQA) is getting popular for requesting valid information from experienced people. However, waiting for such favorable answers for a new submitted question, is a boring task for users once querying to online community forums. IR system can utilize thread in online community forum for question queries. Even so, the appropriate answers are often mixed among snippets of many irrelevant documents, and opening full articles is still required. A post-processing system is needed in order to obtain the most relevant answers. CQA tasks want to address this need, to help user get the most favorable answers by improving IR system results.

SemEval CQA Task 3 is designed to gather some possible solutions, in five coherent subtasks (Nakov et al., 2017). Since some subtasks are related, we focus only on subtask B, with goal to provide a good basis framework for solving problem in other subtasks.

In Task 3 of the previous year, word embeddings obtained with a tool such as word2vec (Mikolov et al., 2013, 2013b) contributed to the best systems for all subtasks. In addition, machine learning based methods were mostly ranked in the top positions for all subtasks. The most popular machine learning approach was SVM for classification, regression and ranking, while neural networks, even though widely used, did not win any subtasks (Nakov et al., 2016).

Most machine learning approaches rely on several similarity features as the basis. Various techniques to compute semantic similarity based on word embeddings, were used by Franco-Salvador et al. (2016), Filice et al. (2016), Mohtarami et al. (2016), Wu and Lan (2016), and Mihaylov and Nakov (2016). Besides, they also used various lexical and semantic similarities including simple match counts on words or n-grams. Specifically, Franco-Salvador et al. (2016), also used nouns and n-grams overlaps, distributed word alignments, knowledge graphs, and common frame.

Interestingly, Mihaylova et al. (2016) used cosine distance between topic pairs, and text distance for SVM learning features, rather than using similarity features. They also implemented other Boolean and Qatar Living Forum users as task specific features.

Filice et al. (2016) constructed many types of similarity based on text pairs, e.g. n-grams of word lemmas, n-grams of POS tags, parse tree, and LCS for SVM learning features. Then they stack the classifiers across subtasks to solve substasks B and C in such a way that utilizes other subtasks' results. This task-specific features seem to be the key success for the team to get the relatively best performance on all English subtasks.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 370–374,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

In this CQA task, we focus on machine learning approaches with a small number of features. We attempt to find an effective way to use word embeddings as the basis of our similarity features. We also make use of the words (lemmas) that are frequent in a thread or small document collection (i.e. the original and the 10 related questions), in the calculation of similarity between sentences. We create several sets of words with different 'word importance levels', from which we derive similarity features for machine learning methods.

The experiment on this 2017 shared task (subtask B) shows good results with respect to MAP scores. Our method also surpasses IR baseline and achieved the 7th position out of 13 teams for the primary submission.

2 System Description

The framework of our system contains three main phases, i.e. (1) pre-processing, (2) feature generation, and (3) training and classification.

2.1 Pre-processing

From each dataset, i.e. development, train and test sets, we extract the questions to form threads for subtask B. Each thread contains one original question (orgQ) and the 10 related questions (relQ). We use the term 'collection of documents' for the thread, which contains questions (each with subject and body[1]) as the documents.

From each collection of documents, we extract all lemmas and select only content words: nouns, verbs, adjectives, named entities, question words, and foreign words. For this need we use lemmatizer, POS tagger and Named Entity Recognizer from Stanford CoreNLP (Manning et al., 2014). We also count each lemma's frequency in each collection of documents for each certain thread, not from the whole dataset.

Intuitively, in a QA forum, if the frequency of a word is high in a certain thread, the word is likely to be an important matter in the conversation discussed by majority users. For this reason, we rank the words by their frequencies. We list top-*N* rank of words[2] for next process. In our experiments, we set *N* to 4.

2.2 Word Importance Level

We first derive several sets of content words from orgQ$_{\text{subj}}$ (the set of words in the subject of orgQ), orgQ$_{\text{body}}$ (the set of words in the body of orgQ), and TopN consisting the top *N* words in the ranking obtained in Section 2.1. Specifically, the following sets are supposed to have different levels of importance:

L1=$orgQ_{sub} \cap TopN$,
L2=$TopN \cap (orgQ_{sub} \cup orgQ_{body})$,
L3=$TopN$,
L4=$orgQ_{sub} \cup TopN$,
L5=$orgQ_{sub} \cup orgQ_{body}$,
L6=$\neg(TopN \cap (orgQ_{sub} \cup orgQ_{body}))$.

For example, the words in L1 belong to both set of orgQ-subject and TopN, and thus supposed to be very important.

2.3 Similarity Feature

We next calculate a number of similarities between two sets of content words: C_{orgQ} representing orgQ such as L1 and L2, and C_{relQ} representing relQ such as relQ$_{\text{sub}}$, relQ$_{\text{body}}$, and their union. We later use these similarities as features for the classifier as in Table 1.

Semantic Similarity

The first semantic similarity type in this work is the cosine similarity (Equation (2)) between the sums (resultant *R* as in Equation (1)) of word embeddings of the words *w* in the sets.

$$R = \sum_{i=1}^{n} w_i \quad (1)$$

$$Sim(C_{orgQ}, C_{relQ}) = cos\theta = \frac{R_{orgQ} \cdot R_{relQ}}{|R_{orgQ}||R_{relQ}|} \quad (2)$$

As word embeddings, we use the pre-trained Google 1B words dataset, with 300-dimensional word vectors (Mikolov et al., 2013b).

Lexical Semantic Similarity

For the second type of similarity, we use lexical semantic similarity, which is similar to Konopık et al. (2016). We denote the union of C_{orgQ} and C_{relQ} by C $(i.e., C = C_{orgQ} \cup C_{relQ}$, which consists of *m* unique words $\{b_1, \ldots, b_m\}$.

Given two sets C_{orgQ} and C_{relQ}, we derive their *m*-dimensional lexical vector representations LV_{orgQ} and LV_{relQ} respectively. For each word b_i in C, we calculate the maximum cosine similar-

[1] If body is empty, we copy the subject for the body.

[2] Only words with frequency count ≥ 2 are taken into consideration.

ity score between the embeddings of b_i and a word in C_{orgQ}, which we regard as an element of LV_{orgQ}:

$$LV_{orgQ} = \left\{ \max_{w \in C_{orgQ}} \left(Sim(b_1, w) \right), \dots, \max_{w \in C_{orgQ}} \left(Sim(b_m, w) \right) \right\}. \tag{3}$$

Similarly, we calculate each element of LV_{relQ} from C_{relQ}. Lastly, we calculate the cosine similarity between LV_{orgQ} and LV_{relQ} to form a new feature.

2.4 Feature Generation

For our supervised learning, we compose feature sets as Table 1 below. Semantic cosine similarity is indexed with i in {1, ..., 10} and lexical semantic similarity with j in {11, ..., 20}.

F_i	F_j	C_{orgQ}	C_{relQ}
F_1	F_{11}	L1	$relQ_{sub}$
F_2	F_{12}	L1	$relQ_{sub} \cup relQ_{body}$
F_3	F_{13}	L3	$relQ_{sub}$
F_4	F_{14}	L3	$relQ_{sub} \cup relQ_{body}$
F_5	F_{15}	L4	$relQ_{sub}$
F_6	F_{16}	L4	$relQ_{sub} \cup relQ_{body}$
F_7	F_{17}	L5	$relQ_{sub} \cup relQ_{body}$
F_8	F_{18}	L6	$relQ_{sub} \cup relQ_{body}$
F_9	F_{19}	L2	$relQ_{sub} \cup relQ_{body}$
F_{10}	F_{20}	NE	$relQ_{sub} \cup relQ_{body}$

Table 1: Similarity Feature Composition

As additional features, we investigated influence of named entities (NE) in F_{10} and F_{20}. We extract only sentences or questions containing NE-words in orgQ subject and body as C_{orgQ}.

2.5 Learning, Classification and Ranking

We use machine learning for relevance classification and ranking tasks on the same feature combinations. We extract gold annotations (i.e., relevance and score) from the training set and compose separate SVM input files for both tasks. We run the training to produce models for both tasks.

For classification task, SVM binary classifier with a linear kernel (Joachims, 1999) is used to assign label on each relQ, relevant (true) or not relevant (false) on the test set. For ranking task, SVM rank (Joachims, 2002) is used to produce scores. The score assigned to each relQ is regarded as rank, where a higher score means more related to the orgQ. Then, we take both results (relevance and score) into a system prediction file.

3 Experiments and Results

3.1 Dataset

We use 2016 Task 3 datasets provided by the organizer[3], i.e. TRAIN-part1, DEV and TEST. We do not use TRAIN-part2 for it is less reliable and contains more noise as informed in the readme-file. We also conduct experiments on TEST-2016 dataset to test our system performance and compare it with the published official scores in Nakov et al. (2016) as seen in Table 4.

3.2 Feature Selection

We create a simple baseline, which uses only a single similarity feature. This baseline only computes semantic cosine similarity of F_7, i.e. using all content words in orgQ and relQ (word importance level L5). For tuning the parameters and seeking the best combination of features, we train SVM with a linear kernel on TRAIN dataset, and applied the model on DEV dataset. We choose two best cost-parameters *C* with specific feature combinations in Table 2.

Features	Feature Description	*C*=1	*C*=100
7	Base (L5)	69.56	69.56
7,8	Base (L5) + L6	69.76	68.23
7,6	Base (L5) + L4	69.73	69.72
7,4	Base (L5) + L3	69.31	70.06
7,9	Base (L5) + L2	71.09	70.37
7,2	Base (L5) + L1	72.06	72.30
7,1,2	L5+ L1* + L1	70.86	72.09
7,1,2,9	L5+ L1* + L1 + L2	72.33	72.50
7,1,2,9, 17,11,12,19	L5+ L1* + L1 + L2 (both similarity types)	72.26	73.10
1-20	All features, both sim types	**74.04**	**73.19**

L1* means the similarity is computed between L1 and relQ subject only.

Table 2: MAP Scores on DEV

The official score for CQA Task is MAP (Mean Average Precision), besides other complementary scores, i.e. Average Recall, MRR (Mean Reciprocal Rank) Precision, Recall, F_1 and Accuracy (Nakov et al., 2016, 2017).

To analyze the influence of each feature, we conducted experiments on many possible combinations as in Table 2. We combine each word important level from the lowest level (i.e. L6, L4, L3, L2, L1), with baseline (L5) and see how it influences the MAP score. Generally, by combining with other single word importance level features,

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

the MAP score is increased. Combined feature set $F_{7,9}$, i.e. word important level L2 (top-*N* words appear in orgQ subject and body) improves the MAP score by about 1 point when compared with single baseline feature L5. Moreover, we get more improvement when baseline is combined with word important level L1, i.e. top-*N* words in orgQ subject only (experiment with feature set $F_{7,2}$).

We are also curious to join more word importance level features, to compute using both similarity types, and to use different content words of relQ, e.g. content words that appear in subject only or in both subject and body. Some interesting results are also reported in Table 2.

When adding the similarity between L1 and relQ subject only ($F_{7,1,2}$), the MAP score slightly decreases for C=100, but decreases by more than 1 point for C=1. Interestingly, adding one more feature from L2 ($F_{7,1,2,9}$), gives the better score than the aforementioned features.

L1 and L2 tend to have higher influence on the MAP score, compared with L3, L4, and L6. When combining with their lexical semantic similarity features ($F_{7,1,2,9,17,11,12,19}$), L1 and L2 increase MAP score for C=100, but a little bit decrease the score for C=1. Considering that each of L3 to L6 has its own contribution to the improvement of the baseline, we incorporate all features and use both similarity types. The results give the two best MAP scores among all our experiments in this parameter tuning and feature selection phase.

3.3 Final Results

For our participation in Subtask B, we use combination of F_1 - F_{20}, and TRAIN-part1 for training. We choose C=1 and C=100, as the primary and contrastive Con-1 respectively. For contrastive Con-2, we use C=1 and join TRAIN-part1+TEST-2016 for training.

Method	MAP	AvgR	MRR	P	R	F_1	Acc
IR	41.85	77.59	46.42	-	-	-	-
Best	**47.22**	82.60	50.07	27.30	94.48	42.37	52.39
Lowest	40.56	76.67	46.33	36.55	53.37	43.39	74.20
Random	29.81	62.65	33.02	18.72	75.46	30.00	34.77
Primary	43.44	77.50	47.03	35.71	67.48	46.71	71.48
Con-1	44.29	78.59	48.97	34.47	68.10	45.77	70.11
Con-2	43.06	76.45	46.22	35.71	67.48	46.71	71.48

Table 3: Final Result on Task B

Our system achieved the 7th position out of 13 teams for the primary submission with MAP score is 43.44. Our contrastive-1 has the best score among our three submissions, i.e. 44.29, which is nearly about 1 point higher than the primary submission.

Method	MAP	AvgR	MRR	P	R	F_1	Acc
IR	74.75	88.30	83.79	-	-	-	-
Best	**76.70**	90.31	83.02	63.53	69.53	66.39	76.57
Lowest	69.04	84.53	79.55	39.53	64.81	49.11	55.29
Random	46.98	67.92	50.96	40.43	32.58	73.82	45.20
Con-1	72.49	87.77	81.95	64.32	58.88	61.43	75.43

Table 4: Experiment on SemEval 2016 subtask B

We also conduct experiment to test our system performance on TEST-2016 dataset. We use model from TRAIN-part1 dataset training with C=100 (our best result as in Table 3, i.e. Constrastive-1). In respect of previous year results, this result achieved the 8th position out of 12 teams, if it is put into the leaderboard. In respect of the scores, our results in the 2017 and 2016 dataset are consistently in the middle range between the top and the lowest MAP score as seen in Table 4.

4 Conclusion and Future Work

As many CQA tasks rely on similarity measure as the basis, utilizing word importance classes in such a way for semantic similarity metrics can increase the MAP score significantly. Taking into consideration the top-n words in a thread, can contribute to find alternative words, which are unseen in the original question.

Our future work is to implement this method as baseline for other subtasks, and later combine with rich features, which involve various task-specific operations to solve the main problem in CQA.

Acknowledgements

First author would like to acknowledge the Indonesia Endowment Fund for Education (LPDP), Ministry of Finance of the Republic of Indonesia who support his study and research in Japan.

We also would like to thank Prof. Manabu Okumura for the generous support on writing this paper.

References

Guoshun Wu and Man Lan. 2016. *ECNU at SemEval-2016 Task 3: Exploring traditional method and deep learning method for question retrieval and answer ranking in community question answering*.

In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval '16, San Diego, CA. https://aclweb.org/anthology/S/S16/S16-1135.pdf

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. *The Stanford CoreNLP Natural Language Processing Toolkit*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.

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 '16, San Diego, CA. https://aclweb.org/anthology/S/S16/S16-1126.pdf

Mitra Mohtarami, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Tao Lei, Kfir Bar, Scott Cyphers, and Jim Glass. 2016. *SLS at SemEval-2016 Task 3: Neuralbased approaches for ranking in community question answering*. In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval '16, San Diego, CA.
https://aclweb.org/anthology/S/S16/S16-1128.pdf

Miloslav Konopík, Ondrej Prazak, David Steinberger, and Tomáš Brychcín. 2016. *UWB at semeval-2016 task 2: Interpretable semantic textual similarity with distributional semantics for chunks*. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, California, June. Association for Computational Linguistics.
https://aclweb.org/anthology/S/S16/S16-1124.pdf

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, 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, June. Association for Computational Linguistics.
https://aclweb.org/anthology/S/S16/S16-1083.pdf

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. *SemEval-2017 Task 3: Community Question Answering. In Pro ceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, Vancouver, Canada, August. Association for Computational Linguistics.

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 '16, San Diego, CA.
https://aclweb.org/anthology/S/S16/S16-1172.pdf

T. Joachims. 2002. *Optimizing Search Engines Using Clickthrough Data*, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.

T. Joachims, 1999 in: *Making large-Scale SVM Learning Practical*. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.

Todor Mihaylov and Preslav Nakov. 2016. *SemanticZ at SemEval-2016 Task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings*. In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval '16, San Diego, CA. https://aclweb.org/anthology/S/S16/S16-1136.pdf

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space*. In Proceedings of Workshop at ICLR, 2013.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. *Distributed Representations of Words and Phrases and their Compositionality*. In Proceedings of NIPS, 2013.

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, CA. https://aclweb.org/anthology/S/S16/S16-1129.pdf