

Question-Comment Similarity in Community Q&A forums

Master's Thesis Presentation

Sandesh C
12CS30041

Under the supervision of:

Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology – Kharagpur

Table of contents

1. Introduction
2. SemEval '16 Task 3 (Subtask A)
3. Gimplse at relevant Literature
4. Approach
5. Paragraph Vector Representation
6. Experimental Setup
7. Results
8. Conclusion

Introduction

Motivation

- Community Q&A forums are seldom moderated and quite open
- Need to automate the process of finding good/relevant comments to questions
- Ranking the comments according to relevance to the corresponding question

This problem is also referred to as **Answer Reranking**.

SemEval '16 Task 3 (Subtask A)

Semantic Evaluation Tasks

- Ongoing series of evaluations of computational semantic analysis systems¹
- **Task 3** deals with semantic comparison for words and text in Community Question & Answering (CQA) domain

In essence, the main CQA task can be defined as follows:

Given (i) a new question and (ii) a large collection of question-comment threads created by a user community, rank the comments that are most useful for answering the new question.

¹<http://alt.qcri.org/semeval2016/>

Task 3 – Subtask A

In our work we limit ourselves to finding Question-Comment similarity, which comes under the purview of *Subtask A* of *SemEval '16 Task 3*.

Subtask A *Given a question from a question-comment thread, rank the comments as per their relevance (similarity) with respect to the question.*

Exploring the Dataset

Category	Train (Part-I)	Train (Part-II)	Train+Dev+Test (from SemEval 2015)	Dev	Test	Total
Questions	1,411	379	2,480+291+319	244	327	5,451
Comments	14,110	3,790	14,893+1,529+1,876	2,440	3,270	41,908
-Good	5,287	1,364	7,418+813+946	818	1,329	17,975
-Bad	6,362	1,777	5,971+544+774	1,209	1,485	18,122
-Potentially	2,461	649	1,504+172+156	413	456	5,811

Table 1: English CQA-QL corpus from SemEval-2017 Task 3 (Subtask A)

Gimplse at relevant Literature

Recent works which attempt to solve the problem of answer re-ranking:

- [Lin and Wang, 2015] treated the answer selection task as a sequence labeling problem and proposed recurrent convolutional neural networks to recognize good comments
- [Zhou et al., 2015] included long-short term memory (LSTM) units in their convolutional neural network to learn the classification sequence for the thread

SemEval '16 Task 3 – Subtask A

- Notably, at SemEval '16 Task 3 – Subtask A provided a great set of Tree Kernel based approaches, that proved to give best results, out performing various LSTM based approaches
- Although the reason for this could be for the lack of substantial data for an LSTM to be trained over

Approaches based on feature engineering

[Mihaylov and Nakov, 2016] on the other hand constructs various semantic and metadata features and trains an SVM to solve the classification problem of comment relevance.

This idea of using various helpful features from metadata and constructing scores for the similarity between the semantics of question and comment text has been adopted in our approach as well owing to its simplicity.

Approach

Neural Approach

- A neural approach to open-domain non-factoid Question-Answering introduced by [Bogdanova and Foster, 2016]
- Question-comment pairs represented as concatenated distributed representation vectors
- Multilayer Perceptrons to compute similarity scores

Despite its simplicity, their work achieved **state-of-the-art** performance on the Yahoo! Answers dataset of manner or How questions [Jansen et al., 2014].

This improved performance was attributed to the use of **paragraph vector** representations instead of averaging over word vectors.

Proposed Architecture

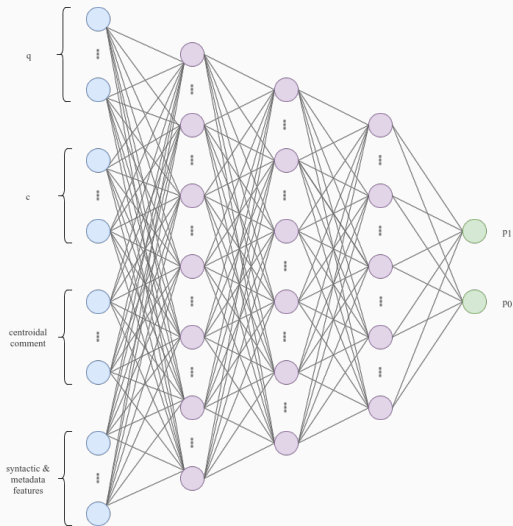


Figure 1: Architecture of proposed Feedforward Neural Network

$$avg_com_q = \frac{\sum_{c \in q} c}{|| \sum_{c \in q} c ||} \quad (1)$$

- Using the information in other comment texts
- Intended to capture the inter-dependency at thread level
- Results in accurate relative relevance scores as output probability is the inverse rank for comment texts

Cosine Similarity

$$\text{sim}(u, v) = 1 - \frac{u \cdot v}{||u|| \cdot ||v||} \quad (2)$$

Centroid Vector

$$\text{centroid}(w_1 \dots w_n) = \frac{\sum_{i=1}^n w_i}{n} \quad (3)$$

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 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 using K-Means clustering. We then calculated the cluster similarity between the question text word clusters and the answer text word clusters.

LDA topic similarity.

- Topic clustering using Latent Dirichlet Allocation (LDA) as implemented in the gensim [Rehurek and Sojka, 2010] toolkit on Train1 + Train2 + Dev questions and comments.
- We built topic models with 100 topics.
- For words in the question text and comment text, we built a bag-of-topics, 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.

Paragraph Vector similarities. The similarity between the distributed vector representations of question text (q), answer text (a) and the centroidal comment (avg_com_q), taken two at a time are also included.

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. Assumption here is that longer answers could bring useful details.

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.

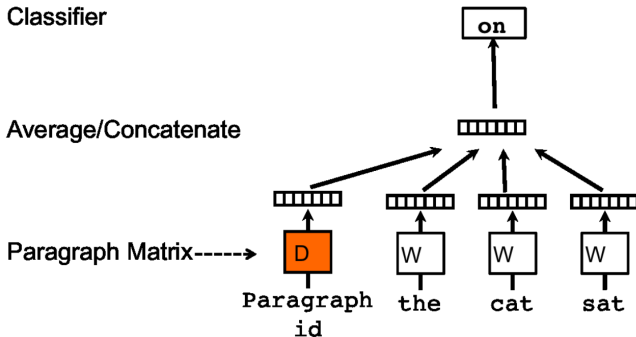
Question category. We took the category of the question as a sparse binary feature vector (a feature with a value of 1 appears if question is in the category). The assumption here is that the question-comment relevance might depend on the category of the question.

Time difference between Question and Comment posting. Immediate comments could reflect incomplete answers to longer questions, while comments posted after substantial time might reflect well-thought answers.

Paragraph Vector Representation

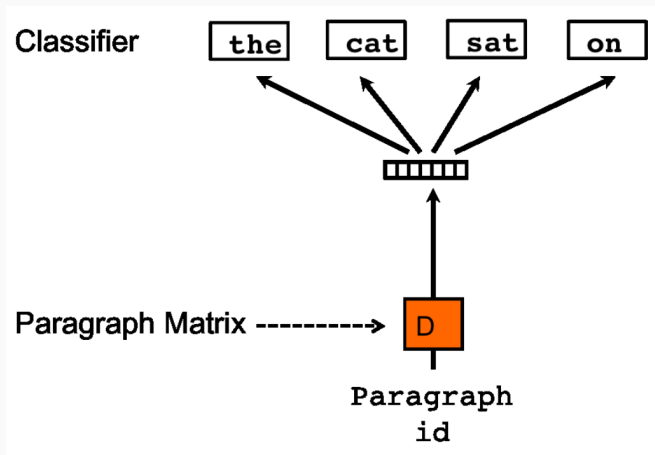
- Paragraph Vector is an unsupervised framework that learns continuous distributed vector representations for variable-length pieces of texts.
- We experimentally evaluate the Paragraph Vector (PV) model proposed by [Le and Mikolov, 2014].
- PV is an extension of the widely used continuous bag-of-words (CBOW) and skip-gram word embedding models, known as word2vec.

Distributed Memory (DM) framework



- Concatenation or average of word vectors with a context of few words is used to predict the next word.
- PV represents the missing information from the current context and can act as a memory of the topic of the paragraph.

Distributed Bag of Words (DBOW) framework



The paragraph vector is trained to predict the words in a small window.

Training Paragraph Vectors

- We use the *gensim*² implementation of DM and DBOW paragraph vector models
- Data for training the unsupervised *doc2vec* model is a large unannotated dataset from Qatar Living forums (~ 2.3 M samples)
- Each piece of text was converted to lowercase, tokenized (by numerics and special characters) and cleaned of stop words

²<https://radimrehurek.com/gensim/models/doc2vec.html>

Framework of PVs – DBOW vs DM

Category	Window Size	Epochs	Normalized Sq. Error (A)	Norm. Sq. Error (Random) (B)	Ratio (B/A)
PV-DBOW	10	5	0.14	0.80	5.89
PV-DBOW	10	10	0.14	0.83	5.84
PV-DM	10	5	0.21	0.99	4.67
PV-DM	15	10	0.22	0.98	4.47

Table 2: Training document vector representations – Best results

- **Window size** is the maximum distance between the predicted word and context words used for prediction within a document
- **Epochs** is the number of iterations for which the *doc2vec* model is trained over dataset
- Note that dimensions of the PVs in these experiments are 100

Dimension of PVs – 100 vs 200

Dimension	Window Size	Epochs	Normalized Sq. Error (A)	Norm. Sq. Error (Random) (B)	Ratio (B/A)
100	10	5	0.14	0.80	5.89
100	10	10	0.14	0.83	5.84
200	10	10	0.15	0.84	5.65
200	10	5	0.15	0.82	5.47

Table 3: Training PV-DBOW vectors of sizes 100 & 200 – Best results

Experimental Setup

Experimental Setup

- For the implementation of the feedforward neural network we used the popular python library *scikit-learn*³'s *MLPClassifier*⁴
- Training data comprises of 38,638 comments over 5,124 questions
- The neural net input is a tuple of the form $(q, c, avg_ans_q, ft_{(q,c)})$
- Nets were trained with parameters being the solvers, activation functions and hidden layer configurations

³<http://scikit-learn.org/stable/index.html>

⁴http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

SemEval Task 3 has as an official evaluation measure used to rank the participating systems, the metric of Mean Average Precision (MAP), calculated for the ten comments a participating system has ranked highest.

Further metrics such as Mean Reciprocal Rank (MRR) and Average Recall ($AvgRec$) for top-10 results; Precision (P), Recall (R), F_1 (with respect to the Good/Relevant class) and Accuracy (Acc) are also reported.

Results

Preliminary experiments with (q, c) inputs

Category	Solver	Activation	MAP	AvgRec	MRR	P	R	F ₁	Acc
PV-DBOW	Adam	logistic	70.49	82.92	77.62	66.01	55.08	60.05	70.21
PV-DBOW	SGD	relu	70.19	82.51	77.16	63.27	59.37	61.26	69.48
PV-DBOW	SGD	logistic	70.18	82.42	77.32	64.12	58.77	61.33	69.88
PV-DBOW	SGD	tanh	70.09	82.45	76.62	63.39	59.14	61.19	69.51
PV-DBOW	Adam	relu	69.93	81.06	76.94	59.98	57.41	58.67	67.13
PV-DBOW	Adam	tanh	69.80	82.31	76.35	63.86	55.46	59.36	69.14
PV-DM	SGD	relu	65.78	78.55	74.58	57.93	53.57	55.67	65.32

Table 4: Preliminary experiments using only (q, c) inputs – Best results

Improvement with inclusion of Centroidal comment

Category	Solver	Activation	MAP	AvgRec	MRR	P	R	F ₁	Acc
PV-DBOW	SGD	relu	73.06	84.16	79.61	66.07	57.86	61.69	70.8
PV-DBOW	SGD	tanh	71.88	83.43	79.11	66.84	56.58	61.29	70.95
PV-DBOW	SGD	logistic	71.79	83.46	79.13	66.52	55.3	60.39	70.52
PV-DBOW	Adam	logistic	71.77	83.42	78.96	66.35	57.26	61.47	70.83
PV-DBOW	Adam	tanh	71.69	83.39	78.64	65.5	57.71	61.36	70.46
PV-DBOW	Adam	relu	71.61	82.63	79.45	62.98	60.05	61.48	69.42

Table 5: Experiments using (q, c, avg_com_q) inputs – Best results

Further improvement with Syntactic and Metadata Features

Category	Solver	Activation	MAP	AvgRec	MRR	P	R	F ₁	Acc
PV-DBOW	SGD	tanh	77.74	88.2	85.58	70.81	63.51	66.96	74.53
PV-DBOW	SGD	relu	77.43	87.96	84.81	70.57	62.6	66.35	74.19
PV-DBOW	SGD	logistic	77.26	87.87	85.51	71.49	63.58	67.3	74.89

Table 6: Experiments using $(q, c, avg_com_q, ft_{(q,c)})$ inputs – Best results

Analysis with $(q, c, ft_{(q,c)})$ inputs

Category	Solver	Activation	MAP	AvgRec	MRR	P	R	F ₁	Acc
PV-DBOW	SGD	logistic	77.22	87.44	84.67	69.62	62.75	66.01	73.73
PV-DBOW	SGD	tanh	77.14	87.44	84.65	69.31	63.21	66.12	73.67
PV-DBOW	SGD	relu	77.03	87.39	84.8	70.32	62.75	66.32	74.10

Table 7: Experiments using $(q, c, ft_{(q,c)})$ inputs – Best results

Analysis with only ($ft_{(q,c)}$) inputs

Category	Solver	Activation	MAP	AvgRec	MRR	P	R	F ₁	Acc
PV-DBOW	SGD	tanh	75.11	86.63	81.78	69.93	59.14	64.08	73.06
PV-DBOW	SGD	relu	74.84	86.25	81.61	70.09	59.59	64.42	73.24
PV-DBOW	SGD	logistic	74.33	85.98	81.35	68.96	59.67	63.98	72.69

Table 8: Experiments using ($ft_{(q,c)}$) inputs – Best results

Conclusion

Summary

Features	MAP	AvgRec	MRR	P	R	F ₁	Acc
(q, c)	70.49	82.92	77.62	66.01	55.08	60.05	70.21
(q, c, avg_com_q)	73.06	84.16	79.61	66.07	57.86	61.69	70.8
$(ft_{(q,c)})$	75.11	86.63	81.78	69.93	59.14	64.08	73.06
$(q, c, ft_{(q,c)})$	77.22	87.44	84.67	69.62	62.75	66.01	73.73
$(q, c, avg_com_q, ft_{(q,c)})$	77.74	88.2	85.58	70.81	63.51	66.96	74.53

Table 9: Best results corresponding to each of the feature sets

Comparing with best submissions at SemEval-2016

Submission	MAP	AvgRec	MRR	P	R	F ₁	Acc
Kelp-primary [Filice et al., 2016]	79.19	88.82	86.42	76.96	55.30	64.36	75.11
ConvKN-contrastive1 [Joty et al., 2016]	78.71	88.98	86.15	77.78	53.72	63.55	74.95
Our Approach	77.74	88.2	85.58	70.81	63.51	66.96	74.53
SUper_team-contrastive1 [Mihaylova et al., 2016]	77.68	88.06	84.76	75.59	55.00	63.68	74.50
ConvKN-primary [Joty et al., 2016]	77.66	88.05	84.93	75.56	58.84	66.16	75.54
SemanticZ-primary [Mihaylov and Nakov, 2016]	77.58	88.14	85.21	74.13	53.05	61.84	73.39
Baseline (chronological)	59.53	72.60	67.83				
Baseline (random)	52.80	66.52	58.71	40.56	74.57	52.55	45.26
Baseline (all true)				40.64	100.00	57.80	40.64
Baseline (all false)							59.36

Table 10: Best submissions and Baselines for SemEval '16 Task 3 – Subtask A

- Our best result has a higher recall value compared to the best submissions at the event
- Infact our method produces the best F_1 score out of all the submissions at the event

- With each inclusion of a new set of features there has been an improvement in the MAP score.
- Further inclusion of features related to syntactic similarity amongst the question, comment text and the centroidal comment, hence is highly likely to be fruitful.
- Using syntax trees of these text pieces to extract more syntactic & semantic similarities is a possible start.

Questions



Bogdanova, D. and Foster, J. (2016).

This is how we do it: Answer reranking for open-domain how questions with paragraph vectors and minimal feature engineering.



Filice, S., Croce, D., Moschitti, A., and Basili, R. (2016).

Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers.


Proceedings of SemEval, 16:1116–1123.



Jansen, P., Surdeanu, M., and Clark, P. (2014).

Discourse complements lexical semantics for non-factoid answer reranking.

In *ACL (1)*, pages 977–986.

-  Joty, S., Moschitti, A., Al Obaidli, F. A., Romeo, S., Tymoshenko, K., and Uva, A. (2016).

Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora.

Proceedings of SemEval, pages 896–903.

-  Le, Q. V. and Mikolov, T. (2014).

Distributed representations of sentences and documents.

In *ICML*, volume 14, pages 1188–1196.

-  Lin, X. Z. B. H. J. and Wang, Y. X. X. (2015).

lcrc-hit: A deep learning based comment sequence labeling system for answer selection challenge.

SemEval-2015, 210.



Mihaylov, T. and Nakov, P. (2016).

Semanticz at semeval-2016 task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings.

Proceedings of SemEval, pages 879–886.



Mihaylova, T., Gencheva, P., Boyanov, M., Yovcheva, I., Mihaylov, T., Hardalov, M., Kiprova, Y., Balchev, D., Koychev, I., Nakov, P., et al. (2016).

Super team at semeval-2016 task 3: Building a feature-rich system for community question answering.

Proceedings of SemEval, pages 836–843.



Rehurek, R. and Sojka, P. (2010).

Software framework for topic modelling with large corpora.

In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.



Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (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*, pages 173–180. Association for Computational Linguistics.



Zhou, X., Hu, B., Chen, Q., Tang, B., and Wang, X. (2015).
**Answer sequence learning with neural networks for answer
selection in community question answering.**
arXiv preprint arXiv:1506.06490.