Large-Scale Goodness Polarity Lexicons for Community Question Answering

Todor Mihaylov
Heidelberg University
AIPHES Research Training Group
Im Neuenheimer Feld 325
Heidelberg, Germany 69120
mihaylov@cl.uni-heidelberg.de

Daniel Balchev,
Yasen Kiprov, Ivan Koychev
Sofia University
5 James Bourchier blvd.
Sofia, Bulgaria 1164
koychev@fmi.uni-sofia.bg

Preslav Nakov
Qatar Computing Research Institute,
HBKU
HBKU Research Complex
Doha, Qatar P.O. Box 5825
pnakov@hbku.edu.qa

ABSTRACT

We transfer a key idea from the field of sentiment analysis to a new domain: community question answering (cQA). The cQA task we are interested in is the following: given a question and a thread of comments, we want to re-rank the comments, so that the ones that are good answers to the question would be ranked higher than the bad ones. We notice that good vs. bad comments use specific vocabulary and that one can often predict the goodness/badness of a comment even ignoring the question, based on the comment contents only. This leads us to the idea to build a good/bad polarity lexicon as an analogy to the positive/negative sentiment polarity lexicons, commonly used in sentiment analysis. In particular, we use pointwise mutual information in order to build large-scale goodness polarity lexicons in a semi-supervised manner starting with a small number of initial seeds. The evaluation results show an improvement of 0.7 MAP points absolute over a very strong baseline, and state-of-the art performance on SemEval-2016 Task 3.

CCS CONCEPTS

•Computing methodologies → Natural language processing; **•Information systems → Question answering;** Retrieval models and ranking; Learning to rank;

KEYWORDS

Community Question Answering, Goodness polarity lexicons, Sentiment Analysis.

ACM Reference format:
Todor Mihaylov, Daniel Balchev,
Yasen Kiprov, Ivan Koychev, and Preslav Nakov. 2017. Large-Scale Goodness Polarity Lexicons
for Community Question Answering. In *Proceedings of SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan, ,* 4 pages.
DOI: 10.1145/3077136.3080757

1 INTRODUCTION

Since the very early days of the field of sentiment analysis, researchers have realized that this task was quite different from other natural language processing (NLP) tasks such as document classification [27], e.g., into categories such as business, sport and politics, and that it crucially needed external knowledge in the form of special sentiment polarity lexicons, which could tell the out-of-context sentiment of some words. See for example the surveys by Pang and Lee [19] and Liu and Zhang [7] for more detail about research in sentiment analysis.

Initially, such sentiment polarity lexicons were manually crafted, and were of small to moderate size, e.g., LIWC [20], General Inquirer [29], Bing Liu's lexicon [4], and MPQA [32], all have 2,000-8,000 words. Early efforts in building them automatically also yielded lexicons of moderate sizes [1, 3].

However, recent results have shown that automatically extracted large-scale lexicons (e.g., with a million entries) offer important performance advantages, as confirmed at shared tasks on Sentiment Analysis on Twitter at SemEval 2013-2017 [11, 15–17, 24–26], where over 40 teams participated four years in a row. Similar observations were made in the Aspect-Based Sentiment Analysis task at SemEval 2014-2016 [21–23]. In both tasks, the winning systems benefited from using massive sentiment polarity lexicons [10, 33]. These large-scale automatic lexicons are typically built using bootstrapping, starting with a small set of seeds of, e.g., 50-60 words, and sometimes even just from emoticons [10]; recent work has argued for larger, domain-specific seeds [5]

Here we transfer the idea from sentiment analysis to a new domain: community question answering (cQA). The cQA task we are interested in is this [12–14]: given a question and a thread of comments, we want to rank the comments, so that the ones that are good answers to the question would be ranked higher than the bad ones. We notice that good vs. bad comments have specific vocabulary and that one can often predict goodness/badness even ignoring the question. This leads us to the idea to build a goodness polarity lexicon as an analogy to the sentiment polarity lexicons.

In particular, we use pointwise mutual information (PMI) to build large-scale lexicons in a semi-supervised manner starting with a small number of seeds. The evaluation results on SemEval-2016 Task 3 [14] show that using these lexicons yields state-of-the art performance, and an improvement of 0.7 MAP points absolute over a very strong baseline.

2 PMI AND STRENGTH OF ASSOCIATION

Pointwise mutual information (PMI) comes from the theory of information: given two random variables A and B, the mutual information of A and B is the "amount of information" (in units such as bits) obtained about the random variable A, through the random variable B [2].

PMI is central to a popular approach for bootstrapping sentiment lexicons proposed by Turney [31]. It starts with a small set of seed positive (e.g., *excellent*) and negative words (e.g., *bad*), and then uses these words to induce sentiment polarity orientation for new words in a large unannotated set of texts. The idea is that words that co-occur in the same text with positive seed words are likely to be positive, while those that tend to co-occur with negative words are likely to be negative. To quantify this intuition, Turney defines the notion of *semantic orientation* (*SO*) for a term w as follows:

$$SO(w) = pmi(w, pos) - pmi(w, neg) \tag{1}$$

where *pos* and *neg* stand for any positive and negative seed word, respectively.

The idea was later used by other researchers, e.g., Mohammad et al. [10] built several lexicons based on PMI between words and seed emotional hashtags, i.e., #happy, #sad, #angry, etc. or positive and negative smileys.

3 GOODNESS POLARITY LEXICON

We use SO to build goodness polarity lexicons for *Good*/*Bad* comments in Community Question Answering forums. Instead of using positive and negative sentiment words as seeds, we start with comments that are manually annotated as *Good* or *Bad* (from SemEval-2016 Task 3 datasets [14]).

From these comments, we extract words that are strongly associated with *Good* or *Bad* comments. Finally, we use these words as seeds to extract even more such words, but this time using bootstrapping with unannotated data.

In sum, unlike in the work above, we do not do pure bootstrapping, but rather we have a semi-supervised approach, which works in two steps.

Step 1: To come up with a list of words that signal *Good*/*Bad* comment, and it is not easy to come up with such words manually, we look for words that are strongly associated with the *Good* vs. *Bad* comments in the annotated training dataset (where comments are marked as *Good* vs. *Bad*), using SO. We then select the top 5% of the words with the most extreme positive/negative values of SO; this corresponds to the most extreme *Good*/*Bad* comment words.

Step 2: We apply the SO again, but this time using the seed words selected in Step 1, to build the final large-scale goodness polarity lexicon, as in the above-described work.

Compared to previous work in sentiment analysis lexicon induction, we do not start with a small set of seed words, but rather with a set of comments annotated as *Good* vs. *Bad*, from which we extract *Good*/*Bad* seed words (using SO). Once we have these seed words, we proceed as is done for sentiment analysis lexicon induction (again using SO).

4 EXPERIMENTS AND EVALUATION

We build a system that uses variety of features and is competitive to the best systems in the SemEval-2016 Task 3 competition; we then augment it with features extracted from our PMI-based goodness polarity lexicon. We train an SVM classifier, where we create a training instance for each question-answer pair. Finally, we rank the comments for a given question based on the SVM score.

4.1 Data

We used the data from SemEval-2016 Task 3, Subtask A [14]. It includes 6,398 training questions with 40,288 comments, plus an unannotated dataset comprising 189,941 questions and 1,894,456 comments. We performed model selection on the development dataset: 244 questions and 2,440 answers. The test dataset from the task, which we used for evaluation, included 329 questions and 3,270 comments.

4.2 Non-lexicon Features

We used several semantic vector similarity and metadata features, which we describe below.

Semantic Word Embeddings. We used semantic word embeddings [8] trained using word2vec[1] on the training data plus the unannotated Qatar Living data that was provided by the task organizers. We also used embeddings pre-trained on GoogleNews [9]. 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 (after excluding the stopwords).

Semantic Vector Similarities. We used various cosine 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 word 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.

Part of speech (POS) based word vector similarities. We performed part of speech tagging using the Stanford tagger [30], 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 cluster similarity. We first clustered the word vectors from the word2vec vocabulary into 1,000 clusters using K-Means clustering, which yielded clusters with about 200 words per cluster on average. We then calculated the cluster similarity between the

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

question body's word clusters and the answer's text word clusters. For all experiments, we used clusters obtained from the word2vec model trained on the QatarLiving data with vector size 100, window size 10, minimum word frequency 5, and skip-gram context size 1.

LDA topic similarity. We performed topic clustering using Latent Dirichlet Allocation (LDA) of the questions and of the 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 we calculated similarity. The assumption here is that if the question and the comment share similar topics, they should be more likely to be relevant with respect to each other.

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

Answer contains a question mark. If the comment contains a 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, but the answer is short, it is typically less relevant.

The answer's author is the same as the question's author. It is generally unlikely that the user who asked the question would later on provide a good answer to his/her own question; rather, if s/he takes part in the discussion, it is typically for other reasons, e.g., to give additional detail, to thanks another user, or to ask additional questions [18].

Answer rank in the thread. The idea is that discussion in the forum tends to diverge from the original question over time.

Question category. We took the category of the question as a sparse binary feature vector. The assumption here is that the question-comment relevance might depend on the category of the question.

4.3 Goodness Polarity Lexicon Features

We bootstrapped a goodness polarity lexicon using PMI as described above. This yielded a lexicon[2] of 41,663 words, including 11,932 *Bad* and 29,731 *Good* words, with corresponding weigths, which describe the stregth of association of a word with *Good* and *Bad* comments: positive and negative weights, respectively. The *Good* and the *Bad* words with most extreme weights are shown in Table 1. We can see that the *Good* words mostly refer to locations, which is expected, e.g., for questions asking where something is located. In contrast, the *Bad* words are mostly typos, names, numbers, and words in a foreign language.

Based on the goodness polarity lexicon, we extracted the following features for a target comment: (*i*) number of *Good* and *Bad* words; (*ii*) number of *Good* (and *Bad*) / number of *Good+Bad* words; (*iii*) sum of the scores of the *Good*, sum of *Bad* words, and sum of teh scores for *Good+Bad* words; (*iv*) the highest score for a *Good* word, and the lowest score for a *Bad* word in the answer.

word	**SO**	**word**	**SO**
hayat	6.917	13228	-5.522
flyover	6.195	tting	-4.999
codaphone	6.148	illusions	-4.976
najada	6.145	bach	-4.849
rizvi	6.107	messiah	-4.566
emadi	5.890	dnm	-4.417
passportdept	5.868	daf	-4.356
omran	5.728	2905	-4.328
condenser	5.698	xppg	-4.313
bookstore	5.688	29658	-4.306
azzu	5.634	scorn	-4.219
5552827	5.634	skamu	-4.053
overalling	5.621	rizk	-4.041
muncipilty	5.538	fiddledeedee	-3.954

Table 1: The words with the biggest and the smallest SO scores from our goodness polarity lexicon.

System	**MAP**	AvgRec	MRR
SemEval 1st	79.19	88.82	86.42
Our, with PMI lexicons	***78.75***	***88.64***	***86.69***
Our, no PMI lexicons	***78.08***	***88.37***	***85.19***
SemEval 2nd	77.66	88.05	84.93
SemEval 3rd	77.58	88.14	85.21
...	...	...	...
Average	73.54	84.61	81.54
...	...	...	...
SemEval 12th (Worst)	62.24	75.41	70.58
Baseline$_{time}$	59.53	72.60	67.83
Baseline$_{rand}$	52.80	66.52	58.71

Table 2: Our results compared to those at SemEval, and to two baselines: chronological and random.

4.4 Results

The evaluation results are shown in Table 2. We can see that our system without goodness polarity lexicons would rank second on MAP and AvgRec, and third on MRR, at SemEval-2016 Task 3. It outperforms a random baseline (Baseline$_{rand}$) and a chronological baseline that assumes that early comments are better than later ones (Baseline$_{time}$) by large margins: by about 19 and 25 MAP points absolute (and similarly for the other two measures). It is also well above the worst and the average systems. I.e., this is a very strong system, and thus it is not easy to improve over it. Yet, adding the goodness lexicon features yields about 0.7 points absolute improvement in MAP; the resulting system would have ranked second on MAP and AvgRec, and first on MRR.

5 CONCLUSION AND FUTURE WORK

We have presented experiments in transferring an idea from sentiment analysis to a new domain: community question answering. In particular, we built a goodness polarity lexicon that can help predict whether a comment is likely to be good or bad, regardless of the question asked. We have shown that using the lexicon yielded a sizeable improvement of 0.7 MAP points absolute over a very strong system, and near state-of the art performance on SemEval-2016 Task 3.

[2] Our goodness polarity lexicon is freely-available in the following URL: https://github.com/dbalchev/models/

In future work, we plan to extend the lexicon with *n*-grams. We are further interested in trying other approaches for building polarity lexicons that go beyond PMI, e.g., using weights in SVM [28]; there was a special task on that topic at SemEval-2016 [6]. We also plan to explore the impact of the quality of the words we use as seeds [5].

ACKNOWLEDGMENTS

The work is supported by the NSF of Bulgaria under Grant No.: DN 02/11/2016 - ITDGate. This work is also 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.

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of LREC-2010* (19-21). Valletta, Malta.

[2] Kenneth Ward Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Comput. Linguist.* 16, 1 (March 1990), 22–29.

[3] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of LREC-2006*. Genoa, Italy, 417–422.

[4] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of KDD-2004*. Seattle, WA, 168–177.

[5] Dame Jovanoski, Veno Pachovski, and Preslav Nakov. 2016. On the Impact of Seed Words on Sentiment Polarity Lexicon Induction. In *Proceedings of COLING-2016*. Osaka, Japan, 1557–1567.

[6] Svetlana Kiritchenko, Saif M Mohammad, and Mohammad Salameh. 2016. SemEval-2016 Task 7: Determining Sentiment Intensity of English and Arabic Phrases. In *Proceedings of SemEval-2016*. San Diego, CA, 42–51.

[7] Bing Liu and Lei Zhang. 2012. A Survey of Opinion Mining and Sentiment Analysis. In *Mining Text Data*, Charu C. Aggarwal and ChengXiang Zhai (Eds.). Springer, 415–463.

[8] 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 SemEval-2016*. San Diego, CA, 804 – 811.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS-2013*. Lake Tahoe, CA, 3111–3119.

[10] Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of SemEval-2013*. Atlanta, GA, 321–327.

[11] Preslav Nakov. 2016. Sentiment Analysis in Twitter: A SemEval Perspective. In *Proceedings of WASSA-2016*. San Diego, CA, 171–172.

[12] 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 SemEval-2017*. Vancouver, Canada, 18–39.

[13] 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 SemEval-2015*. Denver, CO, 269–281.

[14] 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 SemEval-2016*. San Diego, CA, 525–545.

[15] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of SemEval-2016*. San Diego, CA, 1–18.

[16] Preslav Nakov, Sara Rosenthal, Svetlana Kiritchenko, Saif M. Mohammad, Zornitsa Kozareva, Alan Ritter, Veselin Stoyanov, and Xiaodan Zhu. 2016. Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. *Language Resources and Evaluation* 50, 1 (2016), 35–65.

[17] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of SemEval-2013*. Atlanta, GA, 312–320.

[18] 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 SemEval-2015*. Denver, CO, 203–209.

[19] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1–2 (2008), 1–135.

[20] James W. Pennebaker, Martha E. Francis, and Roger J. Booth. 2001. *Linguistic Inquiry and Word Count.* Lawerence Erlbaum Associates, Mahwah, NJ.

[21] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of SemEval-2016*. San Diego, CA, 19–30.

[22] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of SemEval-2015*. Denver, CO.

[23] Maria Pontiki, Harris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings SemEval-2014*. Dublin, Ireland, 27–35.

[24] Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of SemEval-2017*. Vancouver, Canada, 493–509.

[25] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of SemEval-2015*. Denver, CO, 450–462.

[26] Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of SemEval-2014*. Dublin, Ireland, 73–80.

[27] Fabrizio Sebastiani. 2002. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.* 34, 1 (March 2002), 1–47.

[28] Aliaksei Severyn and Alessandro Moschitti. 2015. On the automatic learning of sentiment lexicons. In *Proceedings of NAACL/HLT-2015*. Denver, CO, 1397–1402.

[29] Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. The General Inquirer: A Computer Approach to Content Analysis. (1966).

[30] 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 NAACL/HLT-2003*. Edmonton, Canada, 173–180.

[31] Peter D. Turney. 2002. Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of ACL-2002*. Philadelphia, PA, 417–424.

[32] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP-2005*. Vancouver, Canada, 347–354.

[33] Xiaodan Zhu, Svetlana Kiritchenko, and Saif M. Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of SemEval-2014*. Dublin, Ireland, 437–442.