Exploiting Conversational Features to Detect High-Quality Blog Comments

Nicholas FitzGerald, Giuseppe Carenini, Gabriel Murray and Shafiq Joty

University of British Columbia
{nfitz,carenini,gabrielm,rjoty}@cs.ubc.ca

Abstract. In this work, we present a method for classifying the quality of blog comments using Linear-Chain Conditional Random Fields (CRFs). This approach is found to yield high accuracy on binary classification of high-quality comments, with conversational features contributing strongly to the accuracy. We also present a new corpus of blog data in conversational form, complete with user-generated quality moderation labels from the science and technology news blog Slashdot.

1 Introduction and Background

As the amount of content available on the Internet continues to increase exponentially, the need for tools which can analyze and summarize large amounts of text has become increasingly pronounced. Traditionally, most work on automatic summarization has focused on extractive methods, where representative sentences are chosen from the input corpus ([5]). In contrast, recent work (eg. [10], [2]) has taken an abstractive approach, where information is first extracted from the input corpus, and then expressed through novel sentences created with Natural Language Generation techniques. This approach, though more difficult, has been shown to produce superior summaries in terms of readability and coherence.

Several recent works have focused on summarization of multi-participant conversations ([9], [10]). [10] describes an abstractive summarization system for face-to-face meeting transcripts. The approach is to use a series of classifiers to identify different types of messages in the transcripts; for example, utterances expressing a decision being made, or a positive opinion being expressed. The summarizer then selects a set of messages which maximize a function encompassing information about the sentences in which messages appear, and passes these messages to the NLG system.

In this paper, we present our work on detecting high-quality comments in blogs using CRFs. In future work, this will be combined with classification on other axes—for instance that of the message's rhetorical role (ie. Question, Response, Criticism etc.)—to provide the messages for an abstractive summarization system.

CRFs ([7]) are a discriminative probabilistic model which have gained much popularity in Natural Language Processing and Bio-informatics applications.

One benefit of using linear chain CRFs over more traditional linear classification algorithms is that the sequence of labels is considered. Several works have shown the effectiveness of CRFs on similar Natural Language Processing tasks which involve sequential dependencies ([1], [4]). [11] uses Linear-Chain CRFs to classify summary sentences to create extractive summaries of news articles, showing their effectiveness on this task. [6] test CRFs against two other classifiers (Support Vector Machines and Naive-Bayes) on the task of classifying dialogue acts in live-chat conversations. They also show the usefulness of structural features, which are similar to our conversational features (see Sect. 2.3).

2 Automatic Comment Rating System

2.1 The Slashdot Corpus

We compiled a new corpus comprised of articles and their subsequent user comments from the science and technology news aggregation website Slashdot[1]. This site was chosen for several reasons. Comments on Slashdot are moderated by users of the site, meaning that each comment has a scores from -1 to +5 indicating the total score of moderations assigned, with each moderator able to modify the score of a given comment by +1 or -1. Furthermore, each moderation assigns a classification to the comment: for good comments, the classes are: *Interesting*, *Insightful*, *Informative* and *Funny*. For bad comments, the classes are: *Flamebait*, *Troll*, *Off-Topic* and *Redundant*. Since the goal of this work was to identify high-quality comments, most of our experiments were conducted with comments grouped into *GOOD* and *BAD*.

Slashdot comments are displayed in a threaded conversation-tree type layout. Users can directly reply to a given comment, and their reply will be placed underneath that comment in a nested structure. This conversational structure allows us to use Conversational Features in our classification approach (see Sect. 2.3).

Some comments were not successfully crawled, which meant that some comments in the corpus referred to parent comments which had not been collected. In order to prevent this, comments whose parents were missing were excluded from the corpus. After this cleanup, the collection totalled 425,853 comments on 4320 articles.

2.2 Transformation into Sequences

As mentioned above, Slashdot commenters can reply directly to other comments, forming several tree-like conversation for each article. This creates a problem for our use of Linear-Chain CRFs, which require linear sequences.

In order to solve this problem, each conversation tree is transformed into multiple Threads, one for each leaf-comment in the tree. The Thread is the sequence of comments from the root comment to the leaf comment. Each Thread

[1] http://slashdot.org

is then treated as a separate sequence by the classifier. One consequence of this is that any comment with more than one reply will occur multiple times in the training or testing set. This makes some intuitive sense for training, as comments higher in the conversation tree are likely more important to the conversation as a whole, as the earlier a comment appears in the thread the greater effect it has on the course of conversation down-thread. We describe the process of re-merging these comment threads, and investigate the effect this has on accuracy, in Sect. 3.3.

2.3 Features

Each comment in a given sequence was represented as a series of features. In addition to simple unigram (bag-of-words) features, we experimented with two other classes of features: lexical similarity, and conversational features. These are described below:

Similarity Features Three features were used which capture the lexical similarity between two comments: TF-IDF, LSA ([5]) and Lexical Cohesion([3]). For each comment, each of these three scores was calculated for both the preceding and following comment (0 if there was no comment before or after), giving a total of six similarity features. These features were previously shown in [12] to be useful in the task of topic-modelling in email conversations. However, in contrast to [12], where similarity was calculated between sentences, these metrics were adapted to calculate similarity between entire comments.

Conversational Features The conversational features capture information about the how the comment is situated in the conversation as a whole. The list is as follows:

ThreadIndex The index of the comment in the current thread (starting at 0).

NumReplies The number of child comments replying to this

WordLength and *SentenceLength* The length of this comment in words and sentences, respectively.

AvgReplyWordLength and *AvgReplySentLength* The average length of replies to this comment in words and sentence length.

TotalReplyWordLenth and *TotalReplySentLength* The total length of all replies to this comment in words and sentence length.

2.4 Training

The popular Natural Language Machine Learning toolkit MALLET[2] was used to train the CRF model. A 1000-article subset of the entire Slashdot corpus was divided 90%-10% between the training and testing set. The training set consisted of 93,841 Threads from 900 articles, while the testing set consisted of 10,053 Threads from 100 articles.

[2] http://mallet.cs.umass.edu/index.php

	BAD	GOOD
BAD	5991	1965
GOOD	1426	8814
	P:	0.818
	R:	0.861
	F:	0.839

(a)

	P	R	F
all_good	0.563	1.000	0.720
uni	0.708	0.699	0.703
sim	0.802	**0.900**	**0.848**
conv	**0.818**[3]	0.855	0.836
uni_sim	0.780	0.847	0.812
uni_conv	**0.818**[3]	0.855	0.836
sim_conv	**0.818**[3]	0.855	0.836
uni_sim_conv	**0.818**[3]	0.855	0.836

(b)

	BAD	GOOD
BAD	4160	467
GOOD	862	1090
	P:	0.700
	R:	0.558
	F:	0.621

(c)

Table 1: (a) Confusion matrix for binary classification of comment threads. (b) Results of feature analysis on the 3 feature classes. (c) Confusion matrix for re-merged comment threads.

3 Experimental Results

3.1 Classification

Experiment 1 was to train the CRF using data where the full set of moderation labels had been grouped into *GOOD* comments and *BAD*. The Conditional Random Field classifier was trained on the full set of features presented in Sect. 2.3. The Confusion-Matrix of this experiment is presented in Tab. 1a. We can see that the CRF performs well on this formulation of the task, with a precision of 0.818 and recall of 0.839. This compares very favourably to a baseline of assigning *GOOD* to all comments, which yields a precision score of 0.563. The CRF result also performs favourably against a non-sequential Support Vector Machine classifier (P = .799, R = .773) which confirms the existence of sequential dependencies in this problem.

3.2 Feature Analysis

To investigate the relative importance of the 3-types of features (unigrams, similarity, and conversational) we experiment with training the classifier with different groupings of features. The results of this feature analysis is presented in Tab. 1b. All three sets of features can provide relatively good results by themselves, but the similarity and conversational features greatly out-perform the unigram features. Similarity features have a slight edge in terms of recall and f-score, while the Conversational features provide the edge in precision, seeming to dominate Similarity features when both are used. In fact, the results of this analysis seem to show that whenever the conversational features are used, they dominate the effect of the other features, since all sets of features which include Conversational

[3] These results were not identical, though close enough that precision, recall, and f-score were identical to the third decimal point.

features have the same results as using the Conversational features alone. This would seem to indicate that most relevant factors in deciding the quality of a given comment are conversational in nature, including the number of replies it receives and the nature of those replies. This effect could be reinforced by the fact that comments which have previously been moderated as *GOOD* are more likely to be read by future readers, which will naturally increase the number of comments they receive in reply. However, since the unigram- and, more notably, similarity-features can still perform quite well without use of the conversational features, our method is not overly-dependent on this effect.

3.3 Re-Merging Conversation Trees

As described in Sect. 2.2, conversation trees were decomposed into multiple threads in order to cast the problem in the form of sequence labelling. The result of this is that after classification, each non-leaf thread has been classified multiple times, equal to the number of sub-comments of that comment. These different classifications need not be the same, ie. A given comment might well have been classified as *GOOD* in one sequence and *BAD* in another. We next recombined these sequences, such that there is only one classification per comment. Comments which appeared in multiple sequences, and thus received multiple classifications, were marked *GOOD* if they were classified as *GOOD* at least once (*GOOD* if $|\{c_i \in C : c_i = good\}| \geq 1\}$, where C is the set of classifications of comment i[4].

There are two ways to evaluate the merged classifications. The first way is to reassign the newly-merged classifications back onto the thread sequences. This preserves the proportions of observations in the original experiments, which allows us to determine whether merging has affected the accuracy of classification. Doing so showed that there was no significant effect on the performance of the classifier; precision and recall remained .818 and .861, respectively.

The other method is to look at the comment-level accuracy. This removes duplicates from the data, and gives the overall accuracy for determining the classification of a given comment. The results of this are given in Table 1c. The precision and recall in this measure are significantly lower than in the thread-based measure, which indicates that the classification of "leaf" comments tended to be less accurate than that of non-leaf comments which subsequently appeared in more than one thread. The precision of .700 is still much greater than the baseline of assigning *GOOD* to all comments, which would yield a precision of .297. This indicates that our approach can successfully identify good comments.

4 Conclusion and Future Work

In this work, we have presented an approach to identifying high-quality comments in blog comment conversations. By casting the problem as one of binary

[4] This was compared to similar metrics such as a majority-vote metric (*GOOD* if $|\{c_i \in C : c_i = good\}| \geq |\{c_i \in C : c_i = bad\}|$), and performed the best (though the difference was negligible).

classification, and applying sequence tagging by way of a Linear-Chain Conditional Random Field, were were able to achieve high accuracy. Also presented was a new corpus of blog comments, which will be useful for future research.

Future work will focus on refining our ability to classify comments, and incorporating this into an abstractive summarization system. In order to be useful for this task, it would be preferable to have finer-grained classification than just *GOOD* and *BAD*. Applying our current method to the full range of Slashdot moderation classes yielded low accuracy[5]. Future work will attempt to address these issues.

References

1. Chung G.: Sentence retrieval for abstracts of randomized controlled trials. In: BMC Medical Informatics and Decision Making. 2009, 9:10.
2. FitzGerald N., Carenini G., Ng R.: ASSESS: Abstractive Summarization System for Evaluative Statement Summarization, The Pacific Northwest Regional NLP Workshop (NW-NLP), 2010 (extended abstract), Feb 2010
3. Galley M., McKeown K., Fosler-Lussier E. and Jing H.: Discourse segmentation of multi-party conversation In: ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. 2003; Stroudsburg, PA, USA
4. Hirohata K, Okazaki N, Ananiadou S, Ishizuka M: Identifying Sections in Scientific Abstracts using Conditional Random Fields. In: Proceedings of the Third International Joint Conference on Natural Language Processing. January 2008; Hyderabad, India 2008:381-388.
5. Jurafsky D., Martin J.: Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Upper Saddle River, NJ: Pearson Prentice Hall, 2009. Print.
6. Kim S., Cavedon L., Baldwin T.: Classifying dialogue acts in one-on-one live chats. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10) Cambridge, MA, USA.
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 282289
8. McCallum, A.: MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.
9. Murray G., Carenini G.: Summarizing Spoken and Written Conversations. In: EMNLP 2008, Waikiki, Hawaii.
10. Murray G., Carenini G., Ng R.: Generating Abstracts of Meeting Conversations: A User Study. International Conference on Natural Language Generation, (INLG 2010), 2010
11. Shen D., Sun J., Li H., Yang Q., and Chen Z.: Document Summarization using Conditional Random Fields. In: IJCAI '07: International Joint Conferences on Artificial Intelligence.
12. Joty S., Carenini G., Murray G., and Ng, R.: Exploiting Conversation Structure in Unsupervised Topic Segmentation for Emails. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), MIT, Massachusetts, USA.

[5] A longer version of this paper with the report of this experiment is available from the first author's website.