Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)

Automatically Identifying Good Conversations Online (Yes, They Do Exist!)

Courtney Napoles
Johns Hopkins University
napoles@cs.jhu.edu

Aasish Pappu
Yahoo Research
aasishkp@yahoo-inc.com

Joel Tetreault
Grammarly
joel.tetreault@grammarly.com

Abstract

Online news platforms curate high-quality content for their readers and, in many cases, users can post comments in response. While comment threads routinely contain unproductive banter, insults, or users "shouting" over each other, there are often good discussions buried among the noise. In this paper, we define a new task of identifying "good" conversations, which we call ERICs—Engaging, Respectful, and/or Informative Conversations. Our model successfully identifies ERICs posted in response to online news articles with F_1 = 0.73 and F_1 = 0.91 in debate forums.

Introduction

Internet news outlets serve as both a source of curated content and a venue for users to express their opinions and interact with others. These exchanges often range from vacuous to hateful. However, good discussions *do* occur online, with people expressing different viewpoints and attempting to inform, convince or better understand the other side, but they can get lost among the sea of unconstructive comments. We consider a thread *good* when it consists of an Engaging, Respectful, and/or Informative Conversation (ERIC). An example ERIC and non-ERIC are in Table 1. ERICs are characterized by:

- A respectful exchange of ideas, opinions, and/or information in response to a topic(s).
- Opinions expressed as an attempt to elicit a dialogue.
- Comments that seek to contribute some new information or perspective on the relevant topic.

We hypothesize that identifying and promoting ERICs will cultivate a more civil and constructive atmosphere in online communities and potentially encourage more user participation. This work represents the first step towards that goal.

Recent research aims to improve comment quality by identifying engaging comments (FitzGerald et al. 2011; Backstrom et al. 2013), ranking reddit comments by *karma* (Jaech et al. 2015), filtering inflammatory comments (Lin et al. 2012; Nobata et al. 2016) or trolling (Mihaylov and Nakov 2016; Cheng, Danescu-Niculescu-Mizil, and Leskovec 2015), promoting tolerance (Mukherjee et al. 2013), and measuring controversy (Garimella et al. 2016), but none of these attributes alone is indicative of ERICs. More closely related work have measured the quality of individual comments (FitzGerald et al. 2011) or threads on Slashdot using non-linguistic features (Lee, Yang, and Rim 2014). In the latter, user votes are used as a proxy for thread quality.

This work defines a new problem of classifying the holistic quality of entire comment threads in multi-party dialogues and develops methods to identify them in two domains. We describe a method to predict qualities of a sequence of comments with conditional random fields (CRFs; $F_1 \leq 0.91$), and explore four approaches to classifying ERICs, outlined in the following section. We explore the effect of training data size, whether the data coded by trained or untrained annotators, and perform an ablation study to understand what the model has learned. In the domain of online news comment threads, our best performance is $F_1 = 0.73$, and in another domain of debate forums, performance is nearly perfect ($F_1 = 0.91$).

Table 1: An example ERIC (top) and non-ERIC (bottom).

Tooley: Does anyone else think that the cremation was a bit rushed? (only 3 days after his death) Obviously after that happens, no one else will be able to question the findings of the Medical Examiner. Perhaps that is exactly the point. hmmmm...
Doc: Probably per his religion.
anonymous: True. He seems to have had everthing in place and his family seems not to be In need of money. No matter what the cause, he was and still is great and was and still loved by his many fans.
Kala: the tissue samples from the original autopsy will still be preserved

david: I do not understand the hype about that woman, Sia? I do not understand her, her message, assuming she has one. I just do not understand the stupidity of the "performers" today... I never thought I would say that, but I am at a loss!
Lawrence: That's because your a lunatic!!
myptofvw: Interpretive dance isn't something everyone gets. You have to at least appreciate her vocal quality if you can't get the performance.
Miquel: Different strokes, lunatic. I think song-writer/singer/performance artist Sia is brilliant.
david: She sucks!!

Category	P	R	F_1
Persuasive	0.81	0.84	0.91
Audience	0.80	0.99	0.88
Agreement w/ commenter	0.69	0.85	0.76
Informative*	0.76	0.74	0.75
Mean	0.74	0.78	0.75
Controversial*	0.67	0.64	0.65
Disagreement w/ commenter	0.60	0.68	0.64
Off-topic w/ article	0.62	0.67	0.61
Sentiment*	0.44	0.46	0.43

Table 2: Results of predicting comment label sequences. * indicates CRFs that do worse than ridge regression classifiersw trained on the same features.

Experiments

We take four approaches to classifying ERICs: a pipeline (CRF and binary classification), linear classifier with linguistic and social features, an augmented pipeline that incorporates features from the linear model, and a convolutional neural network. The dataset used is from the Yahoo News Annotated Comments Corpus (YNACC), which contains nearly $140k$ threads posted on Yahoo News articles in April 2016 (Napoles et al. 2017). $2.3k$ of the threads which have been coded by trained and untrained annotators. The YNACC coding scheme labels characteristics of comments and threads that inform whether threads in online news comments are ERICs (indicated in YNACC with the binary *constructive* label). This work uses the YNACC train/development/test sets, which contain 2130, 100, and 100 threads respectively. Test threads are from articles published in a separate month from the others. 1300 threads were annotated by trained coders and have several characteristics of each comment also labeled, and the remaining comments were annotated on Amazon Mechanical Turk.

M1. Pipeline

We hypothesize that the types of comments in a thread inform whether that thread is an ERIC. Therefore, our first approach is a pipeline that predicts the sequence of YNACC labels of each comment in a thread, and those predictions are features in a binary ERIC classifier. There are nine target labels (listed in Table 2), and we train a separate CRF for each with sklearn–crfsuite, using stochastic gradient descent and $\ell 2$ regularization with cross-validation. The features are 300-dimensional comment representations modeled using the gensim implementation of doc2vec (Řehůřek and Sojka 2010) trained over $135k$ uncoded YNACC comments. We test each model on the development set alone, to remain unbiased in future experiments. All models but Sentiment are strong predictors and beat stratified baselines with F_1 ranging from 0.61 to 0.91 (Table 2). Aside from Sentiment, which is a multi-class problem with four classes, the other labels are binary decisions. Agreement and Disagreement are independent from each other, with the negative class of each indicating the absence of (dis)agreement.

For classifying ERICs, we represent each thread with the output of the CRFs, using both the total count of each predicted label and the mean marginal probability of each, and train a ridge regression classifier with scikit-learn. This approach outperforms a random baseline when tested on the development set ($F_1 = 0.62$ compared to 0.58), while ridge regression with the true sequence labels (Pipeline–*oracle*) does better ($F_1 = 0.70$). Results are shown in Table 3.

	Development			*Test*		
Model	**P**	**R**	$\mathbf{F_1}$	**P**	**R**	$\mathbf{F_1}$
Random	**0.71**	0.49	0.58	0.50	0.36	0.42
Pipeline–*oracle*	0.73	0.67	0.70	0.55	0.67	0.60
Pipeline	0.47	1.00	0.64	0.53	0.98	0.69
Linear	0.67	0.69	0.68	0.78	0.64	0.70
Pipeline+	0.67	0.71	0.69	0.77	0.68	0.73
Neural	0.58	**0.79**	0.67	0.61	0.62	0.62

Table 3: Precision, recall, and F_1 score of ERIC classifiers.

BOW ($21k$)	Counts of tokens.
Embeddings (300)	Averaged word embedding values from Google News vectors (Mikolov et al. 2013).
Entity (12)	Counts of named entity types.
Length (2)	Mean # sentences/comment, # tokens/sentence.
Lexicon (6)	# pronouns; agreement and certainty phrases; discourse connectives; and abusive language.
POS ($23k$)	Counts of 1–3-gram POS tags.
Popularity (4)	# thumbs up (TU), # thumbs down (TD), TU + TD, and $\frac{\text{TU}}{\text{TU+TD}}$.
Similarity (8)	Overlap between comment and headline, first comment, previous comment, and all previous comments (if applicable).
User (7)	# comments posted, # threads participated in, # threads initiated, TU and TD received, and commenting rate.

Table 4: Features used in the linear model. The number of features from each group is indicated in parentheses.

M2. Linear model

Next, we select a variety of linguistically-motivated features and statistics about the commenters' behavior to represent each thread (Figure 4). User behavior is calculated in YNACC threads timestamped before the one being represented. Features are extracted from each comment, and a thread is represented by the feature values of the first comment and the mean feature values of all replies (i.e,. each feature has two copies: one for the comment and one for the replies). We train $\ell 1$-regularized logistic regression over the whole training set, selecting the $4k$ best features with ANOVA. This model (Linear) is a better predictor than Pipeline on the development set, and shows just a slight increase in performance on the test set (Table 3).

M3. Pipeline+

We then combine the first two models into Pipeline+, which adds the predicted CRF labels of Pipeline as features in the Linear model. Pipeline+ outperforms Pipeline on both the development and test sets. It slightly exceeds the performance of Linear on the development data and shows a more significant improvement on the test data, with $F_1 = 0.73$.

M4. Neural model

Finally, we train a convolutional neural network (CNN; implemented with Keras and Tensor Flow). Following the model of (Kim 2014), the CNN has an embedding layer initialized with the (Mikolov et al. 2013) vectors and a convolutional layer with filters with window sizes 1–3. Each thread is represented by the mean embeddings of the concatenated comment text. On the development set, this model does nearly as well as Linear ($F_1 = 0.67$), but performance deteriorates on the test set ($F_1 = 0.62$). We are training over just $2.1k$ instances, which is a relatively small amount of data for this type of model. Future work will address the small data size and develop more sophisticated networks.

Analysis

Our best model, Pipeline+, represents characteristics of each comment, linguistic features, and information about the commenters. We evaluate how its performance is effected by altering the size and source of training data. There are $2.3k$ annotated threads in YNACC, and we speculate that more data would help performance. Therefore we systematically train models with an increasing number of training samples, from 100 to 2,130 (Figure 1). More training instances improve the predictive power of the model, however the rate of improvement on the development data slows after training on approximately $1k$ instances. For the test set, the rate of improvement remains fairly constant, suggesting that better performance is possible with more labeled data. This difference is likely due to the time period when the threads were posted within the different data splits: threads in the train and development sets were both posted in the same month, and could have had similar article topics. However, the test threads were posted in a later month, and therefore the article topics may not overlap with the training threads, in which case more training data would be beneficial.

We also compare the goodness of fitting a model to data annotated by just trained or untrained annotators (Figure 1). On the development set, training on data annotated by trained annotators does better than just using untrained annotators and, on the test set, training on untrained annotations outperforms a model trained on slightly more threads annotated by either set. The reason behind this result is uncertain as the test set was coded by just trained annotators.

To understand how different feature groups contribute to the model, we perform an ablation study, where we train $\ell 2$-linear regression models using features from each group individually (Table 5). On the development set, leaving out User features increases performance on the development set to $F_1 = 0.70$ and ablating Lexicon, Popularity, and Similarity features does not decrease performance. The features that contribute the most are BOW and POS on both the development and test sets, and removing either of these feature groups substantially diminishes performance (by 0.07–0.08 on development and 0.08–0.11 on test).

99% of the features chosen in feature selection are POS or BOW. POS n-grams with the greatest negative weights are <PRP$ CD> and <" VBZ ">. <PRP$ CD> describes *my $0.02*, which can be used to introduce or hedge a controversial or disparaging opinion, and the quotation marks in <" VBZ "> imply a degree of incredulity or sarcasm. One of the POS n-grams with the highest positive coefficient is <(DT)>. This is a pattern that frequently occurs in formal news text, and so we may infer that ERICs tend to quote the article.

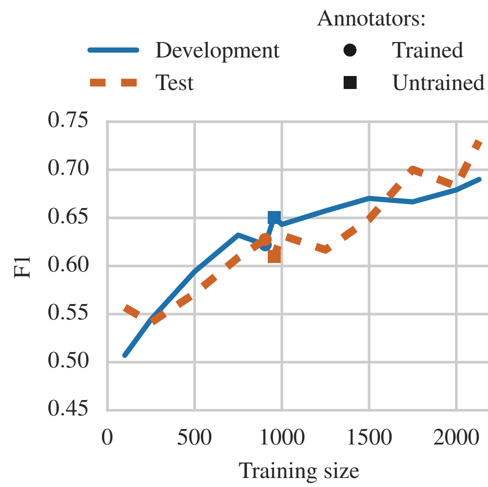

Figure 1: F_1 of predictions using an increasing number of training instances. ● and ■ indicate models trained exclusively on labels from trained and untrained workers.

Feature group	*Development* P	R	F$_1$	*Test* P	R	F$_1$
–BOW	0.64	**0.60**	**0.63**	0.70	0.60	0.65
–CRF	0.67	0.69	0.68	0.78	0.66	0.71
–Embeddings	0.66	0.69	0.67	0.77	0.64	0.70
–Entity	0.67	0.71	0.68	0.75	0.64	0.69
–Length	0.66	0.69	0.67	0.76	0.64	0.69
–Lexicon	0.68	0.71	0.69	0.78	0.66	0.71
–Popularity	0.67	0.71	0.69	0.78	0.64	0.70
–POS	**0.62**	0.68	0.64	**0.66**	**0.58**	**0.62**
–Similarity	0.66	0.71	0.69	0.76	0.62	0.69
–User	0.67	0.73	0.70	0.76	0.64	0.70

Table 5: Results of a feature ablation study.

Turning to the BOW features, the tokens with the greatest negative weight are mostly charged words or words that may occur in a controversial context: *fatal*, *heterosexual*, *grief*, *urinate*, *hostage*, *jews*, and *deporting*. Most of the highest positive weights are given to less controversial words, such as *risk*, *disaster*, *playlist*, and *unattractive*.

Cross-domain Experiments

Finally, we test how well a model that predicts ERICs in the YNACC performs in another domain. The Internet Argument Corpus (IAC) contain threads in which users debate contentious issues (Abbott et al. 2016; Walker et al. 2012), and $1k$ of these have been coded using the same annotations as the Yahoo News (YN) threads in YNACC (Napoles et al. 2017). IAC threads are categorically different from YN in terms of their intent (debate on a particular topic) and significantly longer length. We randomly select 100 IAC threads

Test Set	Model	P	R	F_1
IAC	Baseline	0.78	0.77	0.78
	Pipeline+/YN	0.79	0.75	0.77
	Pipeline+/IAC	0.90	0.93	0.91
	Pipeline+/IAC&YN	**0.91**	**0.93**	**0.92**
YN dev	Pipeline+/YN	**0.67**	**0.69**	**0.68**
	Pipeline+/IAC	0.59	0.64	0.62
	Pipeline+/IAC&YN	0.63	0.71	0.67
YN test	Pipeline+/YN	**0.76**	**0.69**	**0.72**
	Pipeline+/IAC	0.64	0.56	0.60
	Pipeline+/IAC&YN	0.62	0.58	0.60

Table 6: Cross-domain experiments with Yahoo and IAC.

to test with our best model, Pipeline+. A majority class classifier is a very strong baseline on the IAC ($F_1 = 0.78$), and Pipeline+ trained on YN threads (Pipeline+/YN) does not outperform this ($F_1 = 0.77$). If we train Pipeline+ on the IAC data (Pipeline+/IAC), the model has near perfect performance ($F_1 = 0.91$), which is further improved by training on both IAC and YN threads (Pipeline+/IAC&YN).

When using Pipeline+/IAC to test on YN threads, the performance is worse than using the same number of YN training instances on both the development ($F_1 = 0.62$ compared to 0.64) and test sets ($F_1 = 0.60$ versus 0.63). Pipeline+/IAC&YN is a stronger predictor on the YN development set but not as good as the model trained just on YN. Pipeline+/IAC&YN does worse on the YN test set, which may be due to idiosyncrasies in the data, e.g., topics trending when the YN development threads were posted could have overlapped with the IAC debate topics, and not be present in YN test threads. Overall, the presence of out-of-domain (IAC) training data decreases performance on YN threads, however the classification of IAC threads is not hurt by the presence of out-of-domain data.

Conclusion and Future Work

We have identified and defined *ERICs* in online conversations and developed a model to identify them. Even with the broad definition of ERICs, we are able to classify them with $F_1 = 0.73$ in domain and with $F_1 = 0.92$ on out-of-domain threads, using predicted comment labels, a variety of linguistically motivated features, and information about the users. The concept of ERICs can be applied to any user-generated content where users are interacting in an unmoderated venue, such as discussion groups, messaging services, and comments on blogs and microblogs. Future work includes examining the interplay of different comment types and when certain comment types appear in the thread, exploring features such as the relationships between different comment types, time difference between comments, and interactions between different threads (*sub-dialogues*) in a larger dialogue (all threads posted in response to an article).

Acknowledgments

We are grateful to Brian Provenzale and Enrica Rosato for their contributions to this project. We also wish to thank Danielle Lottridge, Smaranda Muresan, Amanda Stent, and the anonymous reviewers for their feedback.

References

Abbott, R.; Ecker, B.; Anand, P.; and Walker, M. A. 2016. Internet Argument Corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it. In *LREC*, 4445–4452.

Backstrom, L.; Kleinberg, J.; Lee, L.; and Danescu-Niculescu-Mizil, C. 2013. Characterizing and curating conversation threads: Expansion, focus, volume, re-entry. In *WSDM*, 13–22.

Cheng, J.; Danescu-Niculescu-Mizil, C.; and Leskovec, J. 2015. Antisocial behavior in online discussion communities. In *ICWSM*, 61–70.

FitzGerald, N.; Carenini, G.; Murray, G.; and Joty, S. 2011. Exploiting conversational features to detect high-quality blog comments. In *AI 2011*, 122–127.

Garimella, K.; De Francisci Morales, G.; Gionis, A.; and Mathioudakis, M. 2016. Quantifying controversy in social media. In *WSDM*, 33–42. ACM.

Jaech, A.; Zayats, V.; Fang, H.; Ostendorf, M.; and Hajishirzi, H. 2015. Talking to the crowd: What do people react to in online discussions? In *EMNLP*, 2026–2031.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, 1746–1751.

Lee, J.-T.; Yang, M.-C.; and Rim, H.-C. 2014. Discovering high-quality threaded discussions in online forums. *Journal of Computer Science and Technology* 29(3):519–531.

Lin, C.; Huang, Z.; Yang, F.; and Zou, Q. 2012. Identify content quality in online social networks. *IET Communications* 6(12):1618–1624.

Mihaylov, T., and Nakov, P. 2016. Hunting for troll comments in news community forums. In *ACL*, 399–405.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Mukherjee, A.; Venkataraman, V.; Liu, B.; and Meraz, S. 2013. Public dialogue: Analysis of tolerance in online discussions. In *ACL*, 1680–1690.

Napoles, C.; Tetreault, J.; Rosato, E.; Provenzale, B.; and Pappu, A. 2017. Finding good conversations online: The Yahoo News Annotated Comments Corpus. In *LAW*.

Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; and Chang, Y. 2016. Abusive language detection in online user content. In *WWW*, 145–153.

Řehůřek, R., and Sojka, P. 2010. Software framework for topic modelling with large corpora. In *LREC*, 45–50.

Walker, M. A.; Tree, J. E. F.; Anand, P.; Abbott, R.; and King, J. 2012. A corpus for research on deliberation and debate. In *LREC*, 812–817.