Something's Brewing!
Early Prediction of Controversy-causing Posts from Discussion Features

Jack Hessel
Cornell University
jhessel@cs.cornell.edu

Lillian Lee
Cornell University
llee@cs.cornell.edu

Abstract

Controversial posts are those that split the preferences of a community, receiving both significant positive and significant negative feedback. Our inclusion of the word "community" here is deliberate: what is controversial to some audiences may not be so to others. Using data from several different communities on reddit.com, we predict the ultimate controversiality of posts, leveraging features drawn from both the textual content and the tree structure of the early comments that initiate the discussion. We find that even when only a handful of comments are available, e.g., the first 5 comments made within 15 minutes of the original post, discussion features often add predictive capacity to strong content-and-rate only baselines. Additional experiments on domain transfer suggest that conversation-structure features often generalize to other communities better than conversation-content features do.

1 Introduction

Controversial content — that which attracts both positive and negative feedback — is not necessarily a bad thing; for instance, bringing up a point that warrants spirited debate can improve community health.[1] But regardless of the nature of the controversy, detecting potentially controversial content can be useful for both community members and community moderators. Ordinary users, and in particular new users, might appreciate being warned that they need to add more nuance or qualification to their earlier posts.[2] Moderators could be alerted that the discussion ensuing from some content might need monitoring. Alternately, they could draw community attention to issues possibly needing resolution: indeed, some sites already provide explicit sorting by controversy.

We consider the controversiality of a piece of content in the context of the community in which it is shared, because what is controversial to some audiences may not be so to others (Chen and Berger, 2013; Jang et al., 2017; Basile et al., 2017). For example, we identify "break up" as a controversial concept in the relationships subreddit (a subreddit is a subcommunity hosted on the Reddit discussion site), but the same topic is associated with a *lack* of controversy in the AskWomen subreddit (where questions are posed for women to answer). Similarly, topics that are controversial in one community may simply not be discussed in another: our analysis identifies "crossfit", a type of workout, as one of the most controversial concepts in the subreddit Fitness.

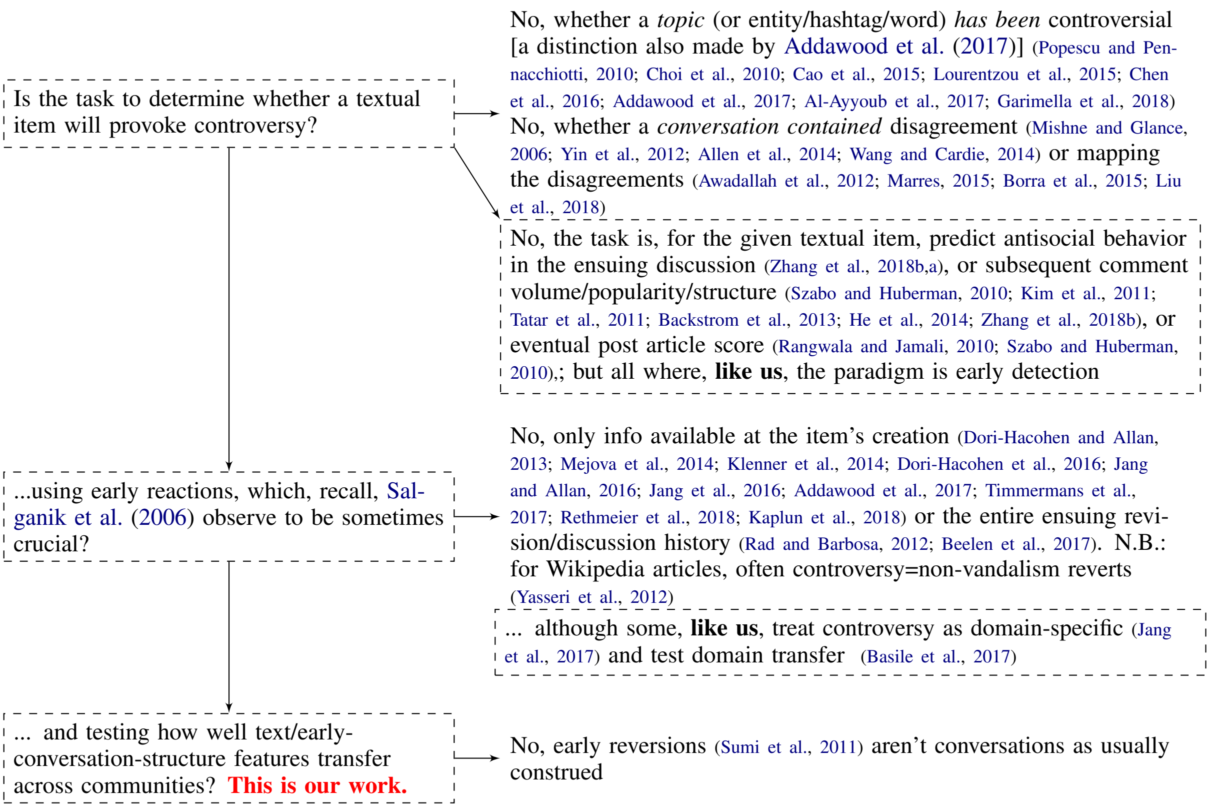
However, while controversial topics may be community-specific, community moderators still may not be able to determine a priori which posts will attract controversy. Many factors cannot be known ahead of time, e.g., a fixed set of topics may not be dynamic enough to handle a sudden current event, or the specific set of users that happen to be online at a given time may react in unpredictable ways. Indeed, experiments have shown that, to a certain extent, the influence of early opinions on subsequent opinion dynamics can override the influence of an item's actual content (Salganik et al., 2006; Wu and Huberman, 2008; Muchnik et al., 2013; Weninger et al., 2015).

Hence, we propose an early-detection approach that uses not just the content of the initiating post, but also the content and structure of the initial responding comments. In doing so, we unite streams of heretofore mostly disjoint research programs: see Figure 1. Working with over 15,000 discus-

[1] Coser (1956); Jehn (1995); De Dreu and Weingart (2003) discuss how disagreement interacts with group makeup, group-task type, and outcome. Chen and Berger (2013) demonstrate a non-linear relationship between controversy and amount of subsequent discussion.

[2] We set aside the issue of *trolls* whose intent is solely to divide a community.

Proceedings of NAACL-HLT 2019, pages 1648–1659
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics



Is the task to determine whether a textual item will provoke controversy?

No, whether a *topic* (or entity/hashtag/word) *has been* controversial [a distinction also made by Addawood et al. (2017)] (Popescu and Pennacchiotti, 2010; Choi et al., 2010; Cao et al., 2015; Lourentzou et al., 2015; Chen et al., 2016; Addawood et al., 2017; Al-Ayyoub et al., 2017; Garimella et al., 2018)
No, whether a *conversation contained* disagreement (Mishne and Glance, 2006; Yin et al., 2012; Allen et al., 2014; Wang and Cardie, 2014) or mapping the disagreements (Awadallah et al., 2012; Marres, 2015; Borra et al., 2015; Liu et al., 2018)

No, the task is, for the given textual item, predict antisocial behavior in the ensuing discussion (Zhang et al., 2018b,a), or subsequent comment volume/popularity/structure (Szabo and Huberman, 2010; Kim et al., 2011; Tatar et al., 2011; Backstrom et al., 2013; He et al., 2014; Zhang et al., 2018b), or eventual post article score (Rangwala and Jamali, 2010; Szabo and Huberman, 2010),; but all where, **like us**, the paradigm is early detection

...using early reactions, which, recall, Salganik et al. (2006) observe to be sometimes crucial?

No, only info available at the item's creation (Dori-Hacohen and Allan, 2013; Mejova et al., 2014; Klenner et al., 2014; Dori-Hacohen et al., 2016; Jang and Allan, 2016; Jang et al., 2016; Addawood et al., 2017; Timmermans et al., 2017; Rethmeier et al., 2018; Kaplun et al., 2018) or the entire ensuing revision/discussion history (Rad and Barbosa, 2012; Beelen et al., 2017). N.B.: for Wikipedia articles, often controversy=non-vandalism reverts (Yasseri et al., 2012)

... although some, **like us**, treat controversy as domain-specific (Jang et al., 2017) and test domain transfer (Basile et al., 2017)

... and testing how well text/early-conversation-structure features transfer across communities? **This is our work.**

No, early reversions (Sumi et al., 2011) aren't conversations as usually construed

Figure 1: How our research relates to prior work.

sion trees across six subreddits, we find that incorporating structural and textual features of budding comment trees improves predictive performance relatively quickly; for example, in one of the communities we consider, adding features taken from just the first 15 minutes of discussion significantly increases prediction performance, even though the average thread only contains 4 comments by that time (∼4% of all eventual comments).

Additionally, we study feature transferability across domains (in our case, communities), training on one subreddit and testing on another. While text features of comments carry the greatest predictive capacity in-domain, we find that discussion-tree and -rate features are less brittle, transferring better between communities.

Our results not only suggest the potential usefulness of granting controversy-prediction algorithms a small observation window to gauge community feedback, but also demonstrate the utility of our expressive feature set for early discussions.

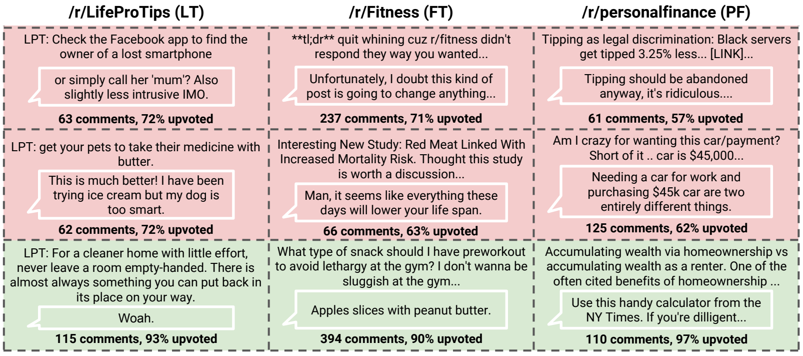
2 Datasets

Given our interest in community-specific controversiality, we draw data from `reddit.com`, which hosts several thousand discussion subcommunities (subreddits) covering a variety of interests. Our dataset, which attempts to cover all public posts and comments from Reddit's inception in 2007 until Feb. 2014, is derived from a combination of Jason Baumgartner's posts and comments sets and our own scraping efforts to fill in dataset gaps. The result is a mostly-complete set of posts alongside associated comment trees.[3] We focus on six text-based[4] subreddits ranging over a variety of styles and topics: two Q&A subreddits: AskMen (AM) and AskWomen (AW); a special-interest community, Fitness (FT); and three advice communities: LifeProTips (LT), personalfinance (PF), and relationships (RL). Each comprises tens of thousands of posts and hundreds of thousands to millions of comments.

In Reddit (similarly to other sites allowing explicit negative feedback, such as YouTube, imgur, 9gag, etc.), users can give posts *upvotes*, increas-

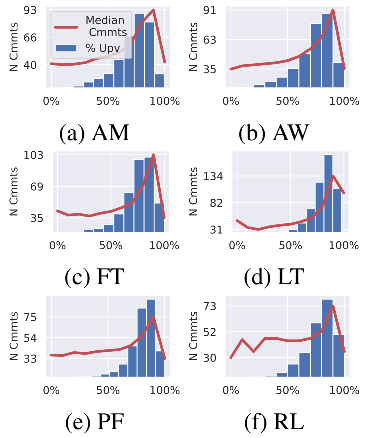
[3] Data hosted at `pushshift.io`, an open data initiative. Scraping was performed using Reddit's API or `github.com/pushshift/api`. Roughly 10% of comments and 20% of posts are deleted by users and/or moderators; also, authorship information is not available for many posts due to deletion of accounts.

[4] We ignore subreddits devoted to image sharing.



/r/LifeProTips (LT)	/r/Fitness (FT)	/r/personalfinance (PF)
LPT: Check the Facebook app to find the owner of a lost smartphone or simply call her 'mum'? Also slightly less intrusive IMO. **63 comments, 72% upvoted**	**tl;dr** quit whining cuz r/fitness didn't respond they way you wanted... Unfortunately, I doubt this kind of post is going to change anything... **237 comments, 71% upvoted**	Tipping as legal discrimination: Black servers get tipped 3.25% less... [LINK]... Tipping should be abandoned anyway, it's ridiculous.... **61 comments, 57% upvoted**
LPT: get your pets to take their medicine with butter. This is much better! I have been trying ice cream but my dog is too smart. **62 comments, 72% upvoted**	Interesting New Study: Red Meat Linked With Increased Mortality Risk. Thought this study is worth a discussion... Man, it seems like everything these days will lower your life span. **66 comments, 63% upvoted**	Am I crazy for wanting this car/payment? Short of it .. car is $45,000... Needing a car for work and purchasing $45k car are two entirely different things. **125 comments, 62% upvoted**
LPT: For a cleaner home with little effort, never leave a room empty-handed. There is almost always something you can put back in its place on your way. Woah. **115 comments, 93% upvoted**	What type of snack should I have preworkout to avoid lethargy at the gym? I don't wanna be sluggish at the gym... Apples slices with peanut butter. **394 comments, 90% upvoted**	Accumulating wealth via homeownership vs accumulating wealth as a renter. One of the often cited benefits of homeownership ... Use this handy calculator from the NY Times. If you're dilligent... **110 comments, 97% upvoted**

Figure 2: Examples of two controversial and one non-controversial post from three communities. Also shown are the text of the first reply, the number of comments the post received, and its percent-upvoted.



(a) AM

(b) AW

(c) FT

(d) LT

(e) PF

(f) RL

Figure 3: For each community, a histogram of percent-upvoted and the median number of comments per bin.

ing a post's score, or *downvotes*, decreasing it.[5] While the semantics of up/down votes may vary based on community (and, indeed, each user may have their own views on what content should be upvoted and what downvoted), in aggregate, posts that split community reaction fundamentally differ from those that produce agreement. Thus, in principle, posts that have unambiguously received both many upvotes *and* many downvotes should be deemed the most controversial.

Percent Upvoted on Reddit. We quantify the relative proportion of upvotes and downvotes on a post using *percent-upvoted*, a measure provided by Reddit that gives an estimate of the percent of all votes on a post that are upvotes. In practice, exact values of percent-upvoted are not directly available; the site adds "vote fuzzing" to fight vote manipulation.[6] To begin with, we first discard posts with fewer than 30 comments.[7] Then, we query for the noisy percent-upvoted from each post ten times using the Reddit API, and take a mean to produce a final estimate.

Post Outcomes. To better understand the interplay between upvotes and downvotes, we first explore the outcomes for posts both in terms of percent-upvoted and the number of comments; doing so on a per-community basis has the potential to surface any subreddit-specific effects. In addition, we compute the median number of comments for posts falling into each bin of the histogram. The resulting plots are given in Figure 3.

In general, posts receive mostly positive feedback in aggregate, though the mean percent-upvoted varies between communities (Table 1). There is also a positive correlation between a post's percent-upvoted and the number of comments it receives. This relationship is unsurprising, given that Reddit displays higher rated posts to more users.

A null hypothesis, which we compare to empirically in our prediction experiments, is that popularity and percent-upvoted simply carry the same information. However, we have reason to doubt this null hypothesis, as quite a few posts receive significant attention despite having a low percent-upvoted (Figure 2).

Assigning Controversy Labels To Posts. We assign binary controversy labels (i.e., relatively controversial vs. relatively non-controversial) to posts according to the following process: first, we discard posts where the observed variability across 10 API queries for percent-upvoted exceeds 5%; in these cases, we assume that there are too few total votes for a stable estimate. Next, we discard posts where neither the observed upvote ratio nor the observed score[8] vary at all; in these cases, we cannot be sure that the upvote ratio is insensitive to the vote fuzzing function.[9] Fi-

[5] Vote timestamps are not publicly available.

[6] Prior to Dec. 2016, vote information was fuzzed according to a different algorithm; however, vote statistics for all posts were recomputed according to a new algorithm that, according to a reddit moderator, can "actually be trusted;" `https://goo.gl/yHWeJp`

[7] The intent is to only consider posts receiving enough community attention for us to reliably compare upvote counts with downvotes. We use number of comments as a proxy for aggregate attention because Reddit does not surface the true number of votes.

[8] A score is the (noised) upvotes minus the downvotes.

[9] We validate our filtration process in a later section by directly comparing to Reddit's rank-by-controversy function.

	# posts	# cmnts	μ_{up} cont	μ_{up} noncont
AM	3.3K	474K	66%	90%
AW	3.0K	417K	67%	91%
FT	3.9K	625K	66%	91%
LT	1.6K	208K	68%	91%
PF	1.0K	95K	72%	92%
RL	2.2K	221K	68%	93%

Table 1: Dataset statistics: number of posts, number of comments, mean percent-upvoted for the controversial and non-controversial classes.

nally, we sort each community's surviving posts by upvote percentage, and discard the small number of posts with percent-upvoted below 50%.[10] The top quartile of posts according to this ranking (i.e., posts with mostly only upvotes) are labeled "non-controversial." The bottom quartile of posts, where the number of downvotes cannot exceed but may approach the number of upvotes, are labeled as "controversial." For each community, this process yields a balanced, labeled set of controversial/non-controversial posts. Table 1 contains the number of posts/comments for each community after the above filtration process, and the percent-upvoted for the controversial/non-controversial sets.

2.1 Quantitative Validation of Labels

Reddit provides a sort-by-controversy function, and we wanted to ensure that our controversy labeling method aligned with this ranking.[11] We contacted Reddit itself, but they were unable to provide details. Hence, we scraped the 1K most controversial posts according to Reddit (1K is the max that Reddit provides) for each community over the past year (as of October 2018). Next, we sampled posts that *did not* appear on Reddit's controversial list in the year prior to October 2018 to create a 1:k ratio sample of Reddit-controversial posts and non-Reddit-controversial posts for $k \in \{1, 2, 3\}$, $k = 3$ being the most difficult setting. Then, we applied the filtering/labeling method described above, and measured how well our process matched Reddit's ranking scheme, i.e., the "controversy" label applied by our method matched the "controversy" label assigned by Reddit.

Our labeling method achieves high precision in identifying controversial/non-controversial posts. While a large proportion of posts are discarded, the labels assigned to surviving posts match those assigned by Reddit with the following F-measures at $k = 3$ (the results for $k = 1, 2$ are higher):[12]

	AM	AW	FT	LT	PF	RL
F-measure	97	96	88	90	94	96

In all cases, the precision for the non-controversial label is perfect, i.e., our filtration method never labeled a Reddit-controversial post as non-controversial. The precision of the controversy label was also high, but imperfect; errors could be a result of, e.g., Reddit's controversy ranking being limited to 1K posts, or using internal data, etc.

2.2 Qualitative Validation of Labels

Figure 2 gives examples of controversial and non-controversial posts from three of the communities we consider, alongside the text of the first comment made in response to those posts.

Topical differences. A priori, we expect that the topical content of posts may be related to how controversial they become (see prior work in Fig. 1). We ran LDA (Blei et al., 2003) with 10 topics on posts from each community independently, and compared the differences in mean topic frequency between controversial and non-controversial posts. We observe community-specific patterns, e.g., in relationships, posts about family (top words in topic: "family parents mom dad") are less controversial than those associated with romantic relationships (top words: "relationship, love, time, life"); in AskWomen, a gender topic ("women men woman male") tends to be associated with more controversy than an advice-seeking topic ("im dont feel ive")

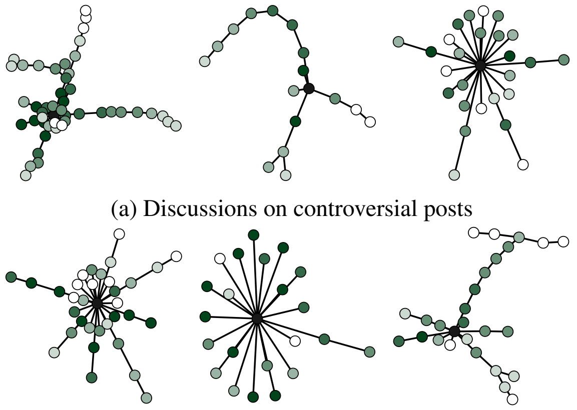
Wording differences. We utilize Monroe et al.'s (2008) algorithm for comparing language usage in two bodies of text; the method places a Dirichlet prior over n-grams (n=1,2,3) and estimates Z-scores on the difference in rate-usage between controversial and non-controversial posts.

This analysis reveals many community-specific patterns, e.g., phrases associated with controversy include "crossfit" in Fitness, "cheated on my" in relationships, etc. What's controversial in one community may be non-controversial in another, e.g., "my parents" is associated with controversy

[10] Reddit provides less information for posts with more upvotes than downvotes.

[11] This validation step rules out the possibility that percent-upvoted is uncorrelated with Reddit's official definition of controversy.

[12] There were communities that we did not consider because the correlation between our filter and Reddit's ranking was lower, e.g., PoliticalDiscussion.



(a) Discussions on controversial posts

(b) Discussions on non-controversial posts

Figure 4: Early conversation trees from AskMen; nodes are comments and edges indicate reply structure. The original post is the black node, and as node colors lighten, comment timing increases from zero minutes to sixty minutes.

in personalfinance (e.g., "live with my parents") but strongly associated with lack of controversy in relationships (e.g., "my parents got divorced"). We also observe that some communities share commonalities in phrasing, e.g., "do you think" is associated with controversy in both AskMen and AskWomen, whereas "what are some" is associated with a lack of controversy in both.

3 Early Discussion Threads

We now analyze comments posted in early discussion threads for controversial vs. non-controversial posts. In this section, we focus on comments posted within one hour of the original submission, although we consider a wider range of times in later experiments.

Comment Text. We mirrored the n-gram analysis conducted in the previous section, but, rather than the text of the original post, focused on the text of comments. Many patterns persist, but the conversational framing changes, e.g., "I cheated" in the *posts* of relationships is mirrored by "you cheated" in the *comments*. Community differences again appear: e.g., "birth control" indicated controversy when it appears in the comments for relationships, but not for AskWomen.

Comment Tree Structure. While prior work in early prediction mostly focuses on measuring rate of early responses, we postulate that more expressive, structural features of conversation trees may also carry predictive capacity.

Figure 4 gives samples of conversation trees that developed on Reddit posts within one hour of the original post being made. There is significant diversity among tree size and shape. To quantify these differences, we introduce two sets of features: C-RATE features, which encode the rate of commenting/number of comments;[13] and C-TREE features, which encode structural aspects of discussion trees.[14] We then examine whether or not tree features correlate with controversy after controlling for popularity.

Using binary logistic regression, after controlling for C-RATE, C-TREE features extracted from comments made within one hour of the original post improve model fit in all cases except for personalfinance ($p < .05$, LL-Ratio test). We repeated the experiment, but also controlled for *eventual* popularity[15] in addition to C-RATE, and observed the same result. This provides evidence that *structural* features of conversation trees are predictive, though *which* tree feature is most important according to these experiments is community-specific. For example, for the models without eventual popularity information, the C-TREE feature with largest coefficient in AskWomen and AskMen was the max-depth ratio, but it was the Wiener index in Fitness.

4 Early Prediction of Controversy

We shift our focus to the task of predicting controversy on Reddit. In general, tools that predict controversy are most useful if they only require information available at the time of submission or as soon as possible thereafter. We note that while the causal relationship between vote totals and comment threads is not entirely clear (e.g., perhaps the comment threads cause more up/down votes on the post), predicting the ultimate *outcome* of posts is still useful for community moderators.

Experimental protocols. All classifiers are bi-

[13] Specifically: total number of comments, the logged time between OP and the first reply, and the average logged parent-child reply time over pairs of comments.

[14] Specifically: max depth/total comment ratio, proportion of comments that were top-level (i.e., made in direct reply to the original post), average node depth, average branching factor, proportion of top-level comments replied to, Gini coefficient of replies to top-level comments (to measure how "clustered" the total discussion is), and Wiener Index of virality (which measures the average pairwise path-length between all nodes in the conversation tree (Wiener, 1947; Goel et al., 2015)).

[15] We added in the logged number of eventual comments, and also whether or not the post received an above-median number of comments.

nary (i.e., controversial vs. non-controversial) and, because the classes are in 50/50 balance, we compare algorithms according to their accuracy. Experiments are conducted as 15-fold cross validation with random 60/20/20 train/dev/test splits, where the splits are drawn to preserve the 50/50 label distribution. For non-neural, feature-based classifiers, we use linear models.[16] For BiLSTM models,[17] we use Tensorflow (Abadi et al., 2015). Whenever a feature is ill-defined (e.g., if it is a comment text feature, but there are no comments at time t) the column mean of the training set for each cross-validation split is substituted. Similarly, if a comment's body is deleted, it is ignored by text processing algorithms. We perform both Wilcoxon signed-rank tests (Demšar, 2006) and two-sided corrected resampled t-tests (Nadeau and Bengio, 2000) to estimate statistical significance, taking the maximum of the two resulting p-values to err on the conservative side and reduce the chance of Type I error.

4.1 Comparing Text Models

The goal of this section is to compare text-only models for classifying controversial vs. non-controversial posts. Algorithms are given access to the full post titles and bodies, unless stated otherwise.

HAND. We consider a number of hand-designed features related to the textual content of posts inspired by Tan et al. (2016).[18]

TFIDF. We encode posts according to tfidf feature vectors. Words are included in the vocabulary if they appear more than 5 times in the corresponding cross-validation split.

W2V. We consider a mean, 300D word2vec (Mikolov et al., 2013) embedding representation, computed from a GoogleNews corpus.

ARORA. A slight modification of W2V, proposed by Arora et al. (2017), serves as a "tough to beat" baseline for sentence representations.

LSTM. We train a Bi-LSTM (Graves and Schmidhuber, 2005) over the first 128 tokens of titles + post text, followed by a mean pooling layer, and then a logistic regression layer. The LSTM's embedding layer is initialized with the same word2vec embeddings used in W2V. Markdown formatting artifacts are discarded.

BERT-LSTM. Recently, features extracted from fixed, pretrained, neural language models have resulted in high performance on a range of language tasks. Following the recommendations of §5.4 of Devlin et al. (2019), we consider representing posts by extracting BERT-Large embeddings computed for the first 128 tokens of titles + post text; we average the final 4 layers of the 24-layer, pretrained Transformer-decoder network (Vaswani et al., 2017). These token-specific vectors are then passed to a Bi-LSTM, a mean pooling layer, and a logistic classification layer. We keep markdown formatting artifacts because BERT's token vocabulary are WordPiece subtokens (Wu et al., 2016), which are able to incorporate arbitrary punctuation without modification.

BERT-MP. Instead of training a Bi-LSTM over BERT features, we mean pool over the first 128 tokens, apply L2 normalization to the resulting representations, reduce to 100 dimensions using PCA,[19] and train a linear classifier on top.

BERT-MP-512. The same as BERT-MP, except the algorithm is given access to 512 tokens (the maximum allowed by BERT-Large) instead of 128.

Results: Table 2 gives the performance of each text classifier for each community. In general, the best performing models are based on the BERT features, though HAND+W2V performs well, too. However, no performance gain is achieved when adding hand designed features to BERT. This may be because BERT's subtokenization scheme incorporates punctuation, link urls, etc., which are similar to the features captured by HAND. Adding an LSTM over BERT features is comparable to mean pooling over the sequence; similarly, considering 128 tokens vs. 512 tokens results in comparable

[16] We cross-validate regularization strength 10^(-100,-5,-4,-3,-2,-1,0,1), model type (SVM vs. Logistic L1 vs. Logistic L2 vs. Logistic L1/L2), and whether or not to apply feature standardization for each feature set and cross-validation split separately. These are trained using `lightning` (`http://contrib.scikit-learn.org/lightning/`).

[17] We optimize using Adam (Kingma and Ba, 2014) with LR=.001 for 20 epochs, apply dropout with $p = .2$, select the model checkpoint that performs best over the validation set, and cross-validate the model's dimension (128 vs. 256) and the number of layers (1 vs. 2) separately for each cross-validation split.

[18] Specifically: for the title and text body separately, length, type-token ratio, rate of first-person pronouns, rate of second-person pronouns, rate of question-marks, rate of capitalization, and Vader sentiment (Hutto and Gilbert, 2014). Combining the post title and post body: number of links, number of Reddit links, number of imgur links, number of sentences, Flesch-Kincaid readability score, rate of italics, rate of boldface, presence of a list, and the rate of word use from 25 Empath wordlists (Fast et al., 2016), which include various categories, such as politeness, swearing, sadness, etc.

[19] Values of 50 and 150 both work well, too.

	AM	AW	FT	LT	PF	RL
HAND	55.4	52.2	61.9	59.7	54.5	60.8
TFIDF	57.4	60.1	63.3	59.1	58.7	65.4
ARORA	58.6	62.0	60.5	59.4	57.2	62.1
W2V	60.7	62.1	63.1	61.4	59.9	64.3
LSTM	58.9	58.2	63.6	61.5	60.0	63.1
BERT-LSTM	**64.5**	**65.1**	**66.2**	65.0	65.1	67.8
BERT-MP	63.4	64.0	64.4	65.7	64.1	67.0
BERT-MP-512	63.9	64.0	64.7	**65.8**	**65.6**	67.7
HAND+W2V	61.3	62.3	64.9	63.2	60.0	66.3
HAND+BERTMP512	63.6	63.5	64.9	64.1	64.4	**68.0**

Table 2: Average accuracy for each post-time, text-only predictor for each dataset, averaged over 15 cross-validation splits; standard errors are ±.6, on average (and never exceed ±1.03). Bold is best in column; underlined are statistically indistinguishable from best in column ($p < .01$)

	AM	AW	FT	LT	PF	RL
TEXT	63.9	64.0	64.7	65.8	65.6	67.7
+TIME	68.1	65.4	65.5	66.2	66.5	69.3
+AUTHOR	68.2	65.3	65.7	66.0	66.4	69.3

Table 3: Post-time only results: the effect of incorporating timing and author identity features.

performance. Based on the results of this experiment, we adopt BERT-MP-512 to represent text in experiments for the rest of this work.

4.2 Post-time Metadata

Many non-content factors can influence community reception of posts, e.g., Hessel et al. (2017) find that *when* a post is made on Reddit can significantly influence its eventual popularity.

TIME. These features encode when a post was created. These include indicator variables for year, month, day-of-week, and hour-of-day.

AUTHOR. We add an indicator variable for each user that appears at least 3 times in the training set, encoding the hypothesis that some users may simply have a greater propensity to post controversial content.

The results of incorporating the metadata features on top of TEXT are given in Table 3. While incorporating TIME features on top of TEXT results in consistent improvements across all communities, incorporating author features on top of TIME+TEXT does not. We adopt our highest performing models, TEXT+TIME, as a strong post-time baseline.

4.3 Early Discussion Features

Basic statistics of early comments. We augment the post-time features with early-discussion feature sets by giving our algorithms access to comments from increasing observation periods. Specifically, we train linear classifiers by combining our best post-time feature set (TEXT+TIME) with features derived from comment trees available after t minutes, and sweep t from $t = 15$ to $t = 180$ minutes in 15 minute intervals.

Figure 6 plots the median number of comments available per thread at different t values for each community. The amount of data available for the early-prediction algorithms to consider varies significantly, e.g., while AskMen threads have a median 10 comments available at 45 minutes, LifeProTips posts do not reach that threshold even after 3 hours, and we thus expect that it will be a harder setting for early prediction. We see, too, that even our maximal 3 hour window is still early in a post's lifecycle, i.e., posts tend to receive significant attention afterwards: only 15% (LT) to 32% (AW) of all eventual comments are available per thread at this time, on average. Figure 7 gives the distribution of the number of comments available for controversial/non-controversial posts on AskWomen at $t = 60$ minutes. As with the other communities we consider, the distribution of number of available posts is not overly-skewed, i.e., most posts in our set (we filtered out posts with less than 30 comments) get at least *some* early comments.

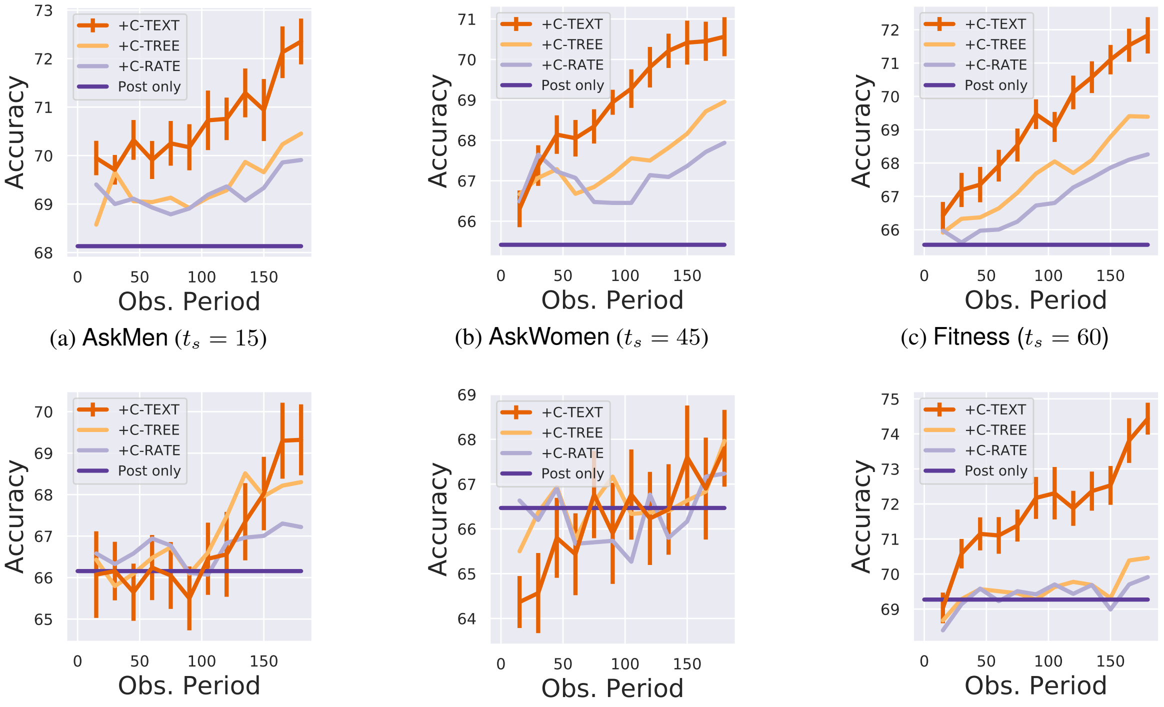
We explore a number of feature sets based on early comment trees (comment feature sets are prefixed with "C-"):

C-RATE and C-TREE. We described these in §3.

C-TEXT. For each comment available at a given observation period, we extract the BERT-MP-512 embedding. Then, for each conversation thread, we take a simple mean over all comment representations. While we tried several more expressive means of encoding the text of posts in comment trees, this simple method proved surprisingly effective.[20]

Sweeping over time. Figure 5 gives the performance of the post-time baseline combined with comment features while sweeping t from 15 to 180 minutes. For five of the six communities we consider, the performance of the comment feature classifier significantly ($p < .05$) ex-

[20] We do not claim that this is the *best* way to represent text in comment trees. However, this simple method produces performance improvements over strong post-time baselines; exploring better models is a promising avenue for future work.



(a) AskMen ($t_s = 15$)

(b) AskWomen ($t_s = 45$)

(c) Fitness ($t_s = 60$)

(d) LifeProTips ($t = 165$)

(e) personalfinance (N/A)

(f) relationships ($t_s = 45$)

Figure 5: Classifier accuracy for increasing periods of observation; the "+" in the legend indicates that a feature set is combined with the feature sets below. t_s, the time the full feature set first achieves statistical significance over the post-time only baseline, is given for each community (if significance is achieved).

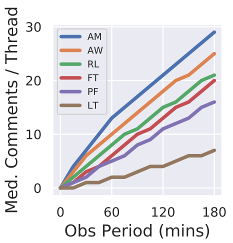


Figure 6: Observation period versus median number of comments available.

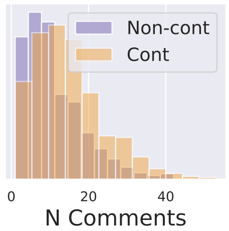


Figure 7: Histogram of the number of comments available per thread at $t = 60$ minutes in AskWomen.

ceeds the performance of the post-time baseline in less than three hours of observation, e.g., in the case of AskMen and AskWomen, significance is achieved within 15 and 45 minutes, respectively.

In general, C-RATE improves only slightly over post only, even though rate features have proven useful in predicting popularity in prior work (He et al., 2014). While adding C-TREE also improves performance, comment textual content is the biggest source of predictive gain. These results demonstrate i) that incorporating a variety of early conversation features, e.g., structural features of trees, can improve performance of controversy prediction over strong post-time baselines, and ii) the text content of comments contains significant complementary information to post text.

Controversy prediction $\neq$ popularity prediction. We return to a null hypothesis introduced in §2: that the controversy prediction models we consider here are merely learning the same patterns that a popularity prediction algorithm would learn. We train popularity prediction algorithms, and then attempt to use them at test-time to predict controversy; under the null hypothesis, we would expect little to no performance degradation when training on these alternate labels.

We 1) train binary popularity predictors using post text/time + comment rate/tree/text features available at $t = 180$,[21] and use them to predict controversy at test-time; and 2) consider an oracle that predicts the true popularity label at test-time; this oracle is *quite* strong, as prior work suggests that perfectly predicting popularity is impossible (Salganik et al., 2006).

[21] We predict whether or not a post eventually receives an above-median number of comments. We force the popularity predictors to predict 50/50 at test time, which improves their performance.

	AM	AW	FT	LT	PF	RL
Pop Pred	53.9	55.2	60.1	54.2	52.9	52.8
Pop Oracle	65.8	67.0	70.3	68.1	64.0	63.3

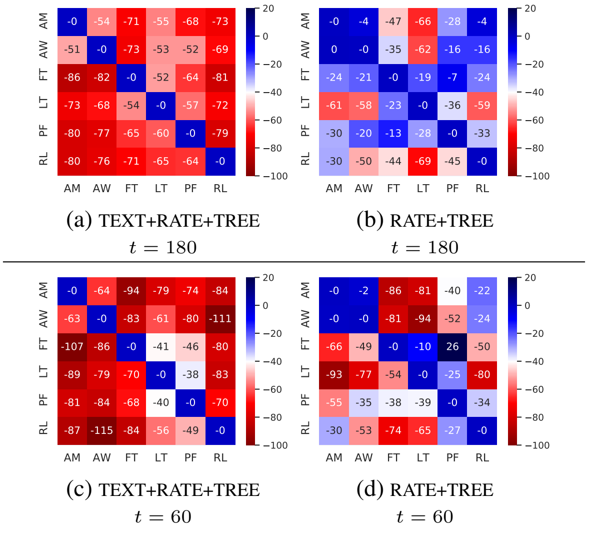
In all cases, the best popularity predictor does not achieve performance comparable to even the post-only baseline. For 3 of 6 communities, even the popularity oracle does not beat post time baseline, and in all cases, the mean performance of the controversy predictor exceeds the oracle by $t = 180$. Thus, in our setting, controversy predictors and popularity predictors learn disjoint patterns.

4.3.1 Domain Transfer

We conduct experiments where we train models on one subreddit and test them on another. For these experiments, we discard all posting time features, and compare C-(TEXT+TREE+RATE) to C-(TREE+RATE); the goal is to empirically examine the hypothesis in §1: that controversial text is community-specific.

To measure performance differences in the domain transfer setting, we compute the percentage accuracy drop relative to a constant prediction baseline when switching the training subreddit from the matching subreddit to a different one. For example, at $t = 60$, we observe that raw accuracy drops from $65.6 \rightarrow 55.8$ when training on AskWomen and testing on AskMen when considering text, rate, and tree features together; given that the constant prediction baseline achieves 50% accuracy, we compute the percent drop in accuracy as: $(55.8 - 50)/(65.6 - 50) - 1 = -63\%$.

The results of this experiment (Figure 8) suggest that while text features are quite strong in-domain, they are brittle and community specific. Conversely, while rate and structural comment tree features do not carry as much in-domain predictive capacity on their own, they generally transfer better between communities, e.g., for RATE+TREE, there is very little performance drop-off when training/testing on AskMen/AskWomen (this holds for all timing cutoffs we considered). Similarly, in the case of training on Fitness and testing on PersonalFinance, we sometimes observe a performance *increase* when switching domains (e.g., at $t = 60$); we suspect that this could be an effect of dataset size, as our Fitness dataset has the most posts of any subreddit we consider, and PersonalFinance has the least.



(a) TEXT+RATE+TREE $t = 180$

(b) RATE+TREE $t = 180$

(c) TEXT+RATE+TREE $t = 60$

(d) RATE+TREE $t = 60$

Figure 8: Average cross-validated performance degradation for transfer learning setting at $t = 180$ and $t = 60$; the y-axis is the training subreddit and the x-axis is testing. For a fixed test subreddit, each column gives the percent accuracy drop when switching from the matching training set to a domain transfer setting. In general, while incorporating comment text features results in higher accuracy overall, comment rate + tree features transfer between communities with less performance degradation.

5 Conclusion

We demonstrated that early discussion features are predictive of eventual controversiality in several reddit communities. This finding was dependent upon considering an expressive feature set of early discussions; to our knowledge, this type of feature set (consisting of text, trees, etc.) hadn't been thoroughly explored in prior early prediction work.

One promising avenue for future work is to examine higher-quality textual representations for conversation trees. While our mean-pooling method did produce high performance, the resulting classifiers do not transfer between domains effectively. Developing a more expressive algorithm (e.g., one that incorporates reply-structure relationships) could boost predictive performance, and enable textual features to be less brittle.

Acknowledgments

We thank C. Danescu-Niculescu-Mizil, J. Zhang, V. Niculae, J. Kleinberg, and the reviewers for helpful feedback, and NVidia Corporation for GPUs. This work was supported in part by NSF grant SES-1741441, but this material does not necessarily reflect the views of the sponsors.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Aseel Addawood, Rezvaneh Rezapour, Omid Abdar, and Jana Diesner. 2017. Telling apart tweets associated with controversial versus non-controversial topics. In *Second Workshop on NLP and Computational Social Science*.

Mahmoud Al-Ayyoub, Abdullateef Rababah, Yaser Jararweh, Mohammed N. Al-Kabi, and Brij B. Gupta. 2017. Studying the controversy in online crowds interactions. *Applied Soft Computing*.

Kelsey Allen, Giuseppe Carenini, and Raymond Ng. 2014. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *EMNLP*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Rawia Awadallah, Maya Ramanath, and Gerhard Weikum. 2012. Harmony and dissonance: organizing the people's voices on political controversies. In *Proceedings of WSDM*, page 523, Seattle, Washington, USA. ACM Press.

Lars Backstrom, Jon Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. 2013. Characterizing and curating conversation threads: expansion, focus, volume, re-entry. In *WSDM*. ACM. Project homepage at `http://www.cs.cornell.edu/home/llee/papers/convcuration.home.html`.

Angelo Basile, Tommaso Caselli, and Malvina Nissim. 2017. Predicting controversial news using Facebook reactions. In *The Italian Conference on Computational Linguistics*.

Kaspar Beelen, Evangelos Kanoulas, and Bob van de Velde. 2017. Detecting Controversies in Online News Media. In *Proceedings of SIGIR*, pages 1069–1072, New York, NY, USA. ACM.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *JMLR*.

Erik Borra, Esther Weltevrede, Paolo Ciuccarelli, Andreas Kaltenbrunner, David Laniado, Giovanni Magni, Michele Mauri, Richard Rogers, and Tommaso Venturini. 2015. Societal Controversies in Wikipedia Articles. In *Proceedings of CHI*, pages 193–196, New York, NY, USA. ACM.

Nan Cao, Lu Lu, Yu-Ru Lin, Fei Wang, and Zhen Wen. 2015. Socialhelix: Visual analysis of sentiment divergence in social media. *Journal of Visualization*.

Wei-Fan Chen, Fang-Yu Lin, and Lun-Wei Ku. 2016. Wordforce: Visualizing controversial words in debates. In *Proceedings of COLING: System Demonstrations*, pages 273–277, Osaka, Japan. The COLING 2016 Organizing Committee.

Zoey Chen and Jonah Berger. 2013. When, why, and how controversy causes conversation. *Journal of Consumer Research*, 40(3).

Yoonjung Choi, Yuchul Jung, and Sung-Hyon Myaeng. 2010. Identifying controversial issues and their subtopics in news articles. In *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer.

Lewis Coser. 1956. *The Functions of Social Conflict*. The Free Press, New York.

Carsten K. W. De Dreu and Laurie R. Weingart. 2003. Task versus relationship conflict, team performance, and team member satisfaction: A meta-analysis.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *JMLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Shiri Dori-Hacohen and James Allan. 2013. Detecting controversy on the web. In *CIKM*. ACM.

Shiri Dori-Hacohen, David Jensen, and James Allan. 2016. Controversy detection in Wikipedia using collective classification. In *Proceedings of SIGIR*, pages 797–800, New York, NY, USA. ACM.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *CHI*.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Quantifying controversy on social media. *ACM Transactions on Social Computing*.

Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J. Watts. 2015. The structural virality of online diffusion. *Management Science*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*.

Xiangnan He, Ming Gao, Min-Yen Kan, Yiqun Liu, and Kazunari Sugiyama. 2014. Predicting the popularity of Web 2.0 items based on user comments. In *SIGIR*.

Jack Hessel, Lillian Lee, and David Mimno. 2017. Cats and captions vs. creators and the clock: Comparing multimodal content to context in predicting relative popularity. In *WWW*. Project homepage at `http://www.cs.cornell.edu/~jhessel/cats/cats.html`.

C.J. Hutto and Eric Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *IWCSM*.

Myungha Jang and James Allan. 2016. Improving automated controversy detection on the web. In *SIGIR*.

Myungha Jang, Shiri Dori-Hacohen, and James Allan. 2017. Modeling controversy within populations. In *SIGIR*.

Myungha Jang, John Foley, Shiri Dori-Hacohen, and James Allan. 2016. Probabilistic approaches to controversy detection. In *CIKM*.

Karen A. Jehn. 1995. A multimethod examination of the benefits and detriments of intragroup conflict. *Administrative Science Quarterly*, 40(2):256–282.

Kateryna Kaplun, Christopher Leberknight, and Anna Feldman. 2018. Controversy and sentiment: An exploratory study. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence (SETN)*, pages 37:1–37:7, New York, NY, USA. ACM.

Su-Do Kim, Sung-Hwan Kim, and Hwan-Gue Cho. 2011. Predicting the virtual temperature of weblog articles as a measurement tool for online popularity. In *11th International Conference on Computer and Information Technology*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Manfred Klenner, Michael Amsler, and Nora Hollenstein. 2014. Verb polarity frames: A new resource and its application in target-specific polarity classification. In *KONVENS*.

Weichen Liu, Sijia Xiao, Jacob T. Browne, Ming Yang, and Steven P. Dow. 2018. ConsensUs: Supporting multi-criteria group decisions by visualizing points of disagreement. *ACM Transactions on Social Computing*, 1(1):1–26.

Ismini Lourentzou, Graham Dyer, Abhishek Sharma, and ChengXiang Zhai. 2015. Hotspots of news articles: Joint mining of news text & social media to discover controversial points in news. In *2015 IEEE International Conference on Big Data*, pages 2948–2950. IEEE.

Noortje Marres. 2015. Why map issues? On controversy analysis as a digital method. *Science, Technology, & Human Values*, 40(5):655–686.

Yelena Mejova, Amy X. Zhang, Nicholas Diakopoulos, and Carlos Castillo. 2014. Controversy and sentiment in online news. In *Computation and Journalism Symposium*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

Gilad Mishne and Natalie Glance. 2006. Leave a reply: An analysis of weblog comments. In *Third Annual Workshop on the Weblogging Ecosystem*. Edinburgh, Scotland.

Burt L. Monroe, Michael P. Colaresi, and Kevin M. Quinn. 2008. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*.

Lev Muchnik, Sinan Aral, and Sean J. Taylor. 2013. Social influence bias: A randomized experiment. *Science*, pages 647–651.

Claude Nadeau and Yoshua Bengio. 2000. Inference for the generalization error. In *NeurIPS*.

Ana-Maria Popescu and Marco Pennacchiotti. 2010. Detecting controversial events from Twitter. In *CIKM*.

Hoda Sepehri Rad and Denilson Barbosa. 2012. Identifying controversial articles in Wikipedia: A comparative study. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration (WikiSym)*, pages 7:1–7:10, New York, NY, USA. ACM.

Huzefa Rangwala and Salman Jamali. 2010. Defining a coparticipation network using comments on Digg. *IEEE Intelligent Systems*.

Nils Rethmeier, Marc Hübner, and Leonhard Hennig. 2018. Learning comment controversy prediction in web discussions using incidentally supervised multitask CNNs. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 316–321.

Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311:854–6.

Róbert Sumi, Taha Yasseri, András Rung, Andrs Kornai, and János Kertész. 2011. Characterization and prediction of Wikipedia edit wars. In *Proceedings of Web Science*. Poster.

Gabor Szabo and Bernardo A. Huberman. 2010. Predicting the popularity of online content. *Comm. of the ACM*.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *WWW*, pages 613–624. Paper homepage at https://chenhaot.com/papers/changemyview.html.

Alexandru Tatar, Jérémie Leguay, Panayotis Antoniadis, Arnaud Limbourg, Marcelo Dias de Amorim, and Serge Fdida. 2011. Predicting the popularity of online articles based on user comments. In *International Conference on Web Intelligence, Mining and Semantics*.

Bram Timmermans, Lora Aroyo, Thomas A. Kuhn, Kaspar Beelen, Evangelos Kanoulas, Bob van de Velde, and Gerben van Eerten. 2017. ControCurator: Understanding controversy using collective intelligence. In *Proceedings of Collective Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Lu Wang and Claire Cardie. 2014. A piece of my mind: A sentiment analysis approach for online dispute detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 693–699, Baltimore, Maryland. Association for Computational Linguistics.

Tim Weninger, Thomas James Johnston, and Maria Glenski. 2015. Random voting effects in social-digital spaces: A case study of Reddit post submissions. In *Conference on Hypertext and Social Media*, pages 293–297, New York, NY, USA. ACM.

Harry Wiener. 1947. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*.

Fang Wu and Bernardo A. Huberman. 2008. How public opinion forms. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, pages 334–341, Berlin, Heidelberg. Springer-Verlag.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. Version 2.

Taha Yasseri, Robert Sumi, Andrs Rung, Andrs Kornai, and Jnos Kertsz. 2012. Dynamics of Conflicts in Wikipedia. *PLoS ONE*, 7(6):e38869.

Jie Yin, Paul Thomas, Nalin Narang, and Cecile Paris. 2012. Unifying local and global agreement and disagreement classification in online debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*. Association for Computational Linguistics.

Justine Zhang, Jonathan P. Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. 2018a. Conversations gone awry: Detecting early signs of conversational failure. In *ACL*. Project homepage at http://www.cs.cornell.edu/~cristian/Conversations_gone_awry.html.

Justine Zhang, Cristian Danescu-Niculescu-Mizil, Christina Sauper, and Sean J. Taylor. 2018b. Characterizing online public discussions through patterns of participant interactions. In *Proceedings of CSCW*. Project homepage at http://www.cs.cornell.edu/~cristian/Patterns_of_participant_interactions.html.