Diversionary Comments under Blog Posts

JING WANG, CLEMENT T. YU, PHILIP S. YU, and BING LIU, University of Illinois at Chicago
WEIYI MENG, SUNY at Binghamton

There has been a recent swell of interest in the analysis of blog comments. However, much of the work focuses on detecting comment spam in the blogsphere. An important issue that has been neglected so far is the identification of diversionary comments. Diversionary comments are defined as comments that divert the topic from the original post. A possible purpose is to distract readers from the original topic and draw attention to a new topic. We categorize diversionary comments into five types based on our observations and propose an effective framework to identify and flag them. To the best of our knowledge, the problem of detecting diversionary comments has not been studied so far. We solve the problem in two different ways: (i) rank all comments in descending order of being diversionary and (ii) consider it as a classification problem. Our evaluation on 4,179 comments under 40 different blog posts from Digg and Reddit shows that the proposed method achieves the high mean average precision of 91.9% when the problem is considered as a ranking problem and 84.9% of F-measure as a classification problem. Sensitivity analysis indicates that the effectiveness of the method is stable under different parameter settings.

CCS Concepts: • **Information systems** → *Spam detection*; • **Mathematics of computing** → *Bayesian networks;*

Additional Key Words and Phrases: Diversionary comments, spam, topic model, latent Dirichlet allocation, hierarchical Dirichlet process, coreference resolution, extraction from Wikipedia, ranking, classification

ACM Reference Format:
Jing Wang, Clement T. Yu, Philip S. Yu, Bing Liu, and Weiyi Meng. 2015. Diversionary comments under blog posts. ACM Trans. Web 9, 4, Article 18 (September 2015), 34 pages.
DOI: http://dx.doi.org/10.1145/2789211

1. INTRODUCTION

Blogs, as a type of Web-based publications consisting of periodic posts with user comments, have been extensively used by individuals to express their views since the late 1990s. According to statistics published in Tumblr[1] and WordPress[2] in February 2014, there were approximately 247.8 million existing blogs, and the population was predicted to double roughly every 5.5 months [Bhattarai et al. 2009]. With such a rapid growth, the number of comments under blog posts also proliferates proportionally. As

[1]https://www.tumblr.com/about.
[2]https://wordpress.com/.

This work was partially supported by the National Science Foundation through grant CNS-1115234, grant IIS-1407927, a Google Research Award, and the Pinnacle Lab at Singapore Management University. The article is an extension of the paper that appeared in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*.
Authors' addresses: J. Wang, C. T. Yu, P. S. Yu, and B. Liu, Dept of Computer Science, University of Illinois at Chicago, IL, 60607, USA; emails: {jwang69, cyu, psyu}@uic.edu, liub@cs.uic.edu; W. Meng, Dept of Computer Science, Binghamton University, NY, 13902, USA; email: meng@binghamton.edu.

DOI: http://dx.doi.org/10.1145/2789211

a strong force of public opinions, blog comments attract attention from people with different backgrounds. Ideally, commentators write their truthful opinions to help shape and build the contents in the blog posts. However, in practice, various types of unrelated comments are deliberately written. For instance, merchants design a simple agent to automatically write unrelated comments with hyperlinks to boost the ranks of target Web pages, companies post advertisements to promote products, and trolls leave off-topic comments to upset people or start arguments. In addition, regular commentators can also change their concerns and intentions subconsciously as the discussion continues, which leads to diversionary comments from the original post.

Many kinds of unrelated comments in the blogosphere have drawn interests from researchers. One type of unrelated comments has hyperlinks to commercially oriented pages and is defined as comment spam [Bhattarai et al. 2009]. It is basically a form of Web spam that aims to mislead search engines to obtain a higher-than-deserved ranking. Various initiatives have been taken to reduce comment spam. Mishne [2005] developed language models for the blog post, post comments, and pages linked by comments to classify comments based on the disagreement among the three models. Based on the features that spam comments are usually shorter than legitimate comments, Cormack et al. [2007] conducted their work on filtering of short messages. Their idea is to improve the traditional bag-of-words spam filter by adding more features such as word bigrams separated by three or fewer words, character bigrams, and character trigrams. However, we did not find any study on detecting uncommercially oriented comments that try to divert to another topic. Based on a study of 12,583 comments for 135 blog posts from Digg[3] and Reddit,[4] we only observed a small percentage of comments containing hyperlinks (4.5%), but we found a significantly higher percentage of comments shifting the discussion topic (30.7%). One example of these diversionary comments is that given a post suggesting that users locking and encrypting their smartphones when most commentators write to share the methods about protecting phones, a diversionary comment changes the topic to discuss why people care more about banking stuff than personal information. And starting from this point, the following comments become more diverse and even divert the topic to government policy in the United States and Canada, including spying networks and health care systems. When readers of this post want to get more ideas about how to encrypt their phones, surely nondiversionary comments are of interest.

In this article, we define comments diverting the discussion from the original post as *diversionary comments*. Based on our observation, we categorize diversionary comments into the following five types (the type distribution among diversionary comments is also given based on a manually labeled dataset of 4,179 comments for 40 randomly chosen blog posts):

—*Type 1 (60.2%; comments shifting to different topics)*. Those that change the discussion topic to another one. It may appear in one of the following forms:
  (1) The blog post discusses topic x, and a diversionary comment directly changes the topic to y, which is somewhat unrelated to x under the blog context, although they might be related under a much broader context, especially under the political issues. The "social security" diversionary comment under the post about cutting defense spending provides an example for this subtype. By extracting the discussion topics in this kind of diversionary comments, we can use them to recommend users for further reading.

[3]http://digg.com/.
[4]http://www.reddit.com/.

(2) The blog post discusses topic x; an earlier comment talked about both x and y, where x and y are unrelated; and a diversionary comment continues on topic y without mentioning topic x. Consider a blog post about Facebook popularity decline among teens, in which an earlier commentator claimed that he never had a Facebook account or owned a television. A diversionary comment proceeds on the topic about television: "Actually the not having a TV thing is becoming a popular trend. I'd join but I'm too much of a gamer to abandon owning a TV, despite rarely watching TV."

—*Type 2 (22.6%; comments about personal attack to commentators)*. Those that comment on the behavior of some preceding commentators without discussing anything related to the topic of the original blog post. An example of this type of diversionary comment is "What's the matter with you? Are you only posting at the very lowest level of threads so you don't deal with responses?"

—*Type 3 (9.7%; comments with little content)*. Those that lack content and only contain words such as "lol" and "hahaha." Even though they might express agreements or disagreements with the preceding commentators or the content of the blog post, their relatedness to the post content is not clear, and therefore they are considered as diversions.

—*Type 4 (4.7%; comments about the hosting website only)*. Those that complain or commend the blog hosting website. We consider them as unrelated to the post content. An example diversionary comment of this type is "Everyone should boycott Digg on Monday." In this comment, Digg is the hosting website.

—*Type 5 (2.7%; advertisements)*. Those that introduce products or refer to companies or websites, and all of which are unrelated to the post content.

Based on the preceding observations, we report a study of identifying diversionary comments. We propose a framework to solve this problem in two different ways, depending on whether the final step applies a ranking algorithm or a classification algorithm. The two approaches use the same set of features. Whereas the ranking algorithm takes the features as scores to rank comments in descending order of being diversionary, the classification algorithm takes those features to build a classifier.

In the post-comments environment, each comment either replies to the post or to a preceding comment. The basis to recognize a legitimate comment is that it is either highly related to the post content or closely related to the preceding comment it replies to, with respect to the topics discussed in the post. In contrast, a diversionary comment is related neither to the post nor to its reply-to comments with respect to the topics in the post content (a comment's reply-to comment is the one it replies to). Relatedness between two documents (a document is either a post or a comment) can be measured by some form of similarity. Consider a post and its comments. The post is usually much longer than an ordinary comment. As a consequence, the proportion of terms or topics in common between the post (P) and a comment (C) is usually not larger than that between a comment (C) and its reply-to comment (RC). So a normalized similarity between P and C is usually smaller than that between C and RC. If a threshold t is set to decide whether a comment is highly related to the post, then an even higher threshold than t should be set to measure the high relatedness between a comment and its reply-to comment. Our method tries to first represent each comment and the post by a vector, then to use a similarity function to compute the relatedness between each comment and the post, and that between each comment and the comment it replies to. Finally, we rank comments based on the similarity scores or classify comments by using these similarity scores as features. However, the following reasons make this a challenging task:

(1) It is difficult to find an accurate representation for each comment and the post. Comments are relatively short and can only offer limited literal information. A

simplistic way of applying term frequencies to build document vectors would yield low accuracies, because a related comment may not share enough words with the post, whereas a diversionary comment may share significant words with the post. For instance, given a post with the topic of President Obama's accomplishments, a diversionary comment that doubts Obama's birthplace shares the significant word "Obama" with the post.

(2) Pronouns and hidden knowledge in the comments and post are other obstacles to accurate representations. First, many commentators use pronouns to represent the person or the issue mentioned in the post. Without mapping pronouns to their corresponding proper nouns or phrases, the number of occurrences of the person or issue cannot be captured accurately. Second, comments often mention some proper nouns, including celebrities, product names, company names, and abbreviations that are not explicitly mentioned in the post but are closely related to the post content. For example, when a post discusses policies of Democrats, a related comment may mention President Obama's domestic policy since he represents the Democrats. Without including such knowledge into the comment and the post, they cannot be represented appropriately either. Third, many words or phrases, although different, may refer to the same topics. Thus, when two comments contain different words but refer to the same topics, their representations are different but ideally should be similar.

(3) A commentator may write to reply to the post directly but may also write to follow a preceding comment. Most blog hosting websites offer a reply-to hierarchy for commentators. However, many comments do not follow the hierarchy, which makes it difficult to find to which post a comment replies.

The main contributions of this article are as follows:

(1) It proposes the new problem of identifying diversionary comments and makes the first attempt to solve the problem in the blogosphere.

(2) It introduces several rules to accurately locate the comment to which a comment replies. An effective rule is also proposed to determine whether a comment replies to the post directly.

(3) It proposes an effective approach to identify diversionary comments, which first applies coreference resolution [Bengtson and Roth 2008] to replace pronouns with corresponding proper nouns or phrases; extracts related information from Wikipedia [Gabrilovich and Markovitch 2007] for proper nouns in comments and the post; utilizes the topic modeling method [Blei et al. 2003; Teh et al. 2004] to group related terms into the same topics and represent comments and the post by their topic distributions; and then, according to their similarities to the post and the comments to which they reply, classifies comments or ranks comments in the descending order of being diversionary.

(4) A dataset, which consists of 4,179 comments under 40 different blog posts from Digg.com and Reddit, was annotated by five annotators with substantial agreement. Experiments based on the dataset are performed to verify the effectiveness of the proposed approach versus various baseline methods. The proposed method achieves 91.9% in mean average precision (MAP) [Baeza-Yates and Ribeiro-Neto 2008] when the ranking algorithm is applied and 84.9% in F-measure when the classification algorithm is applied. In addition, its effectiveness remains high under different parameter settings.

2. MOTIVATION

The existence of diversionary comments is a double-edged sword because they not only bring diversification but also noise. On one hand, many blog posts have too many

comments, and readers do not have time to read them all. When readers are only interested in reading strictly on-topic information, diversionary comments are better skipped. As an example, suppose that an investigator wants to examine blog posts and comments related to human trafficking. An effective search engine should return posts and their comments related to this topic while filtering out comments unrelated to the topic so that the investigator can concentrate on the on-topic comments. On the other hand, some diversionary comments reflect commentators' divergent thinking. Although diversionary comments are unrelated to the original post, some readers may still find some of them interesting. In other words, this diversionary topic could be recommended to the other readers of this post. For example, under a post about "the risky rush to cut defense spending," a diversionary comment changes the topic to social security. Although social security is not strictly related to defense spending cut under the context, both social security and defense are important government programs and their budgets are somewhat related. Thus, some readers of this post might also be interested in the topic about social security. In addition, by identifying diversionary comments, we can provide a high-level summary of discussion topics in comments and let the readers learn the major shifts of discussion topics. This is desirable, as it helps to solve the information overload problem and enables readers to focus. Furthermore, identifying diversionary comments across different blog posts can also help to identify biased commentators or trolls. If a commentator is found to write diversionary comments frequently, then he or she is more likely to be affected by ideologies or be a troll who intends to harm discussions. In summary, a commentator who writes a diversionary comment may deliberately mislead other readers to a different topic or try to broaden the topic under consideration. Irrespective of the intentions of the authors of diversionary comments, the identification of diversionary comments is desirable.

We believe that the problem is also of interest to social networks. Facebook is building a system that tries automatically to block irrelevant comments. According to news from TechCrunch,[5] when a well-known tech startup enthusiast tried to post a comment under a Facebook post about the nature of today's tech blogging scene, he received an error message from Facebook: "This comment seems irrelevant or inappropriate and can't be posted. To avoid having comments blocked, please make sure they contribute to the post in a positive way." However, his comment itself was just expressing agreement with the post and adding in his own ideas. Later a Facebook spokesperson explained that his comment received a "false positive" as spam and stated that Facebook built this automated system to maintain a trusted environment. Clearly, Facebook believes that an automated system that can block off-topic comments is important. However, it is critical for such a system to be highly accurate to reduce complaints from users. Therefore, developing an effective method to identify diversionary comments is very important.

We also conduct a user study to verify the effect of identifying diversionary comments. In this study, we randomly pick 20 blog posts and draw a set of their associated diversionary comments (there are 448 such comments) from our labeled dataset. Each participant in this study is provided a few blog posts and a set of their associated comments. For each comment, the participant is asked the following question:

Assume that you hold interest in the post discussion topic, is this comment of interest to you?

Interestingly, the results show that 83.7% of diversionary comments are of no interest to the participants. The diversionary comments that draw participants' interests mostly (93.2%) belong to subtype 1 of type 1 and may connect to the blog post under a much

[5]http://techcrunch.com/2012/05/05/facebooks-positive-comment-policy-irrelevant-inappropriate-censorship/.

broader context. Given that a diversionary comment is likely to be of no interest to the readers, we believe that identifying diversionary comments provides readers an option to quickly find the comments of interest, and therefore improves the readers' experience. In this work, we flag diversionary comments so that readers can decide whether to read diversionary comments or skip them.

3. RELATED WORK

By analyzing different types of diversionary comments, we realize that types 2, 3, and 5 belong to the traditional spam in different contexts. Therefore, we discuss related work on various types of spam detection. We are not aware of any work on detecting type 1 and type 4 diversionary comments.

The most investigated types of spam are Web spam [Castillo et al. 2006; Castillo and Davison 2010; Martinez-Romo and Araujo 2009; Ntoulas et al. 2006; Wang et al. 2007] and email spam [Blanzieri and Bryl 2008; Cormack 2008; Twining et al. 2004; Zhuang et al. 2008]. Web spam can be classified into content spam and link spam. Content spam involves adding irrelevant words in pages to fool search engines. In the environment of our study, the commentators do not add irrelevant words, as they want to keep their comments readable. Link spam is the spam of hyperlinks, but as we discussed in the previous section, diversionary comments seldom contain hyperlinks. Email spam targets individual users with direct mail messages and are usually sent as unsolicited and nearly identical commercial advertisements to many recipients. Spam emails are filtered based on recurrent features such as the use of some specific words. However, diversionary comments are mostly not commercially oriented and may not contain the same kind of features. In addition, comments are written within the context of the post and preceding comments, whereas emails are written independently.

Comment spam in the blogosphere has also been studied extensively [Bhattarai et al. 2009; Mishne 2005; Cormack et al. 2007; Sculley and Wachman 2007]. It is actually a form of Web spam but is written under a post. There are some important differences between such spam and diversionary comments. First, comment spam typically contains hyperlinks to external pages, whereas diversionary comments seldom do, and therefore techniques that involve using the information from hyperlinks cannot be applied to identify diversionary comments. Second, comment spam is relatively short compared to legitimate comments and usually repeats the same words in a certain pattern to attract search engines, whereas most diversionary comments have similar lengths as those related comments and rarely repeat the same words. Therefore, techniques based on features of traditional comment spam will not perform effectively for identifying diversionary comments.

Another related type of research surrounds opinion spam detection, although it is not conducted in the blogosphere. Jindal et al. regard untruthful or fake reviews aiming at promoting or demoting products as opinion spam [Jindal and Liu 2008; Jindal et al. 2010]. They tried to identify untruthful opinions, reviews on brands only, and nonreviews as three types of opinion spam. They detected the last two types of spam reviews based on supervised learning and manually labeled examples. They detected untruthful reviews by using duplicate and near-duplicate reviews as the training data, which they believed were likely to be untruthful. However, we are not aware of many duplicate comments under the same post or across different blog posts. In addition, the observed duplicate comments are usually due to accidental resubmissions by the same users. Ott et al. [2011] focus their study on deceptive opinion spam, which is defined as fictitious opinions that have been deliberately written to sound authentic. They developed their approach by integrating work from psychology and computational linguistics. According to them, the best performance of detecting deceptive opinion spam could be reached by the classifier built on linear SVM with features extracted

from the Linguistic Inquiry and Word Count (LIWC) software [Ireland et al. 2007] and the combination of unigrams and bigrams. LIWC software is a popular automated text analysis tool used widely to detect personality traits [Mairesse et al. 2007] and analyze deception. Under the broad area of social science, this method could be applied to the detection of diversionary comments, so we will compare its performance to our proposed approach in the experiment section. However, theoretically, diversionary comments are not identical to untruthful and deceptive reviews, because they are different concepts. Diversionary comments are not necessarily fake or deceptive. In addition, diversionary comments are detected based on the context, which contains both the blog post and the preceding comments, whereas opinion spam detection does not involve the preceding reviews context.

We also investigate related work on extracting semantics from short texts including microblog posts and news feeds, as comments are short texts, and finding diversionary comments requires understanding the semantics of comments and blog posts. There are two popular and effective approaches that have been used extensively. The first approach is to apply the topic model LDA and its variants to learn topics from the texts, whereas the second approach is to augment short text information by adding external knowledge, such as Wikipedia concepts [Gabrilovich and Markovitch 2007]. Our approach has combined both of them and is demonstrated to be more effective than each of the two individual methods. The comparison between these two approaches and our proposed method will be made in the experimental section. Hong and Davison [2010] conduct their work on Twitter. They propose several schemes to group data and then use these different corpora to train the LDA model separately and compare their performance. Yano et al. [2009] propose a CommentLDA model to predict responses to political blog posts that jointly describe the contents of posts, the authors, and the contents of comments. In our work, we group all of the comments under one blog post and build a topic model for them. Tsagkias et al. [2011] try to retrieve related social utterances for a given news articles. They extract external knowledge from the social media utterances that are explicitly linked to the given news to help build a better query. Our proposed method also utilizes external knowledge by retrieving Web pages that are related to the title of the post. Meij et al. [2012] explore their work on adding semantics into microblogs. They try to capture the semantics by automatically identifying concepts that are defined in Wikipedia. Banerjee et al. [2007] also explore the information in Wikipedia to enrich the news or blog feeds. Hu et al. [2009] cluster similar short texts (snippets) by using internal semantics and external semantics. They present a framework to incorporate the internal semantics by parsing texts into segment level, phrase level, and word level, and the external semantics by deriving knowledge from Wikipedia and WordNet. In our work, we also enrich the contents of comments and posts by adding concepts from Wikipedia. In addition, we utilize coreference resolution to resolve pronouns. Overall, our method has combined several techniques to enrich the contents of comments and blog posts, which will be further detailed in Section 5.1.

4. ANALYSIS OF COMMENTS

Before discussing the proposed techniques for identifying diversionary comments, let us first describe the data used in this work and illustrate some data features. Here, we use posts and their comments from Digg v4.0[6] and Reddit.

A standard hierarchy of post comments in these two websites is illustrated in Figure 1. Each comment consists of four features (username, written time, comment level,

[6]Digg v4.0 is the version from July 2, 2010, to July 12, 2012.

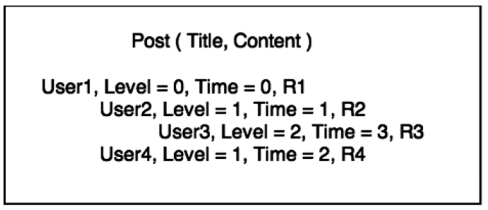

Fig. 1. A snapshot of a hierarchy of a blog post and its comments.

comment content).[7] Among these features, "username" is the commentator's name, whereas "written time" represents the time when the comment is posted. Comments with a comment level of $(n+1)$ are designed to reply to preceding comments of level n. In addition, if a comment's level is 0, then it is supposed to reply to the post content directly. In Figure 1, R1 is a comment of level 0, and R2 is a comment of level 1 that follows the topic of R1. Similarly, R3, with the level of 2, replies to R2.

Under such a hierarchy, we believe that a relevant comment is the one that is either related to the post content directly or related to the preceding comment it replies to, whereas a diversionary comment is unrelated to both the post content and the comment it replies to, with respect to the topics discussed in the post content. Therefore, finding what a comment replies to is necessary for the identification of diversionary comments.

There is existing literature [Aumayr et al. 2011; Zhu et al. 2008; Wang et al. 2011a, 2011b] on finding the reply structure under online forums. The post-comments hierarchy is similar to yet different from the forum reply structure. Comments under a post can respond to the post directly or reply to some previous comments, whereas in a forum thread, people post to reply to previous posts. Aumayr et al. [2011] studied the reply structure of threads in online forums. They extract content and noncontent features and apply the decision tree algorithm to build a classification approach for their task. Among their features, "quotes" (a post quoted a previous post's username, ID, and text section) are a very strong feature in their paper; however, they cannot be applied to our work. Comments under blog posts usually do not quote previous comments' content. Moreover, the "level" feature is distinct under the post-comments structure. In the following, we extract features including username, level, time difference, and content similarities, and provide a set of heuristic rules to effectively detect what a comment replies to.

4.1. Finding What a Comment Replies To

In most cases, a comment at level 0 replies to the blog post content, and a comment at level $(n+1)$ replies to a comment at level n. However, in practice, not all commentators follow such rules to write comments. In Figure 2, $R2$ replies to $R1$ about where to watch political discussions, but it does not follow the standard rule. Therefore, besides the feature of level, we need to combine other features such as written time and username to locate a comment's reply-to comment. We use the following heuristics to find a comment's potential reply-to comments. Assume that comment A is at level n and written at time t, whereas its reply-to comment is written at time t':

(1) If comment A's content contains the information about a username such as "@usernamej," then among comments that precede comment A and are written by "usernamej," the reply-to comment of A is the one that has the smallest positive value of $(t-t')$.

[7]There are other features such as digg numbers and bury numbers for Digg; however, we do not use them, and thus they are not listed here.

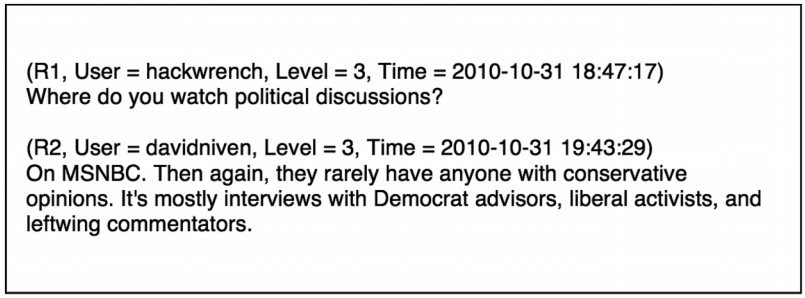

Fig. 2. Comments example.

(2) Among all comments that precede comment A and have the level $(n-1)$, the reply-to comment of A may be the one that has the smallest positive value of $(t - t')$.
(3) Among all comments that precede comment A and have the level n, the reply-to comment of A may be the one that has the smallest positive value of $(t - t')$.
(4) Among all comments that precede comment A, the reply-to comment of A may be the one that has the smallest positive value of $(t - t')$, no matter the level.
(5) If comment B satisfies condition (1), then B is A's reply-to comment; otherwise, all comments that satisfy any of conditions (2), (3), or (4) are considered as potential reply-to comments. If there is only one potential reply-to comment, we consider it as the final reply-to comment. However, if there are multiple potential reply-to comments, we compare the similarities between the comment and all of its potential reply-to comments. Then we choose the one that has the largest similarity based on our method (to be described in Section 5.2).

However, some comments reply to the blog post content directly instead of to other comments. The first comment of the post definitely replies to the post. For each of the other comments at the level of 0, when its similarity to the post is greater than its similarity to its potential reply-to comments, and greater than a specified threshold t (specified as t_3 in Algorithm 1 in Section 5.1.6), we consider it replying to the post directly.

5. DIVERSIONARY COMMENTS IDENTIFICATION

In this section, we present the proposed techniques to identify diversionary comments. We first explain each strategy used to exploit the hidden knowledge and the algorithm used to rank comments. We then discuss the pipeline of our method.

5.1. Techniques

As we mentioned in the previous section, a diversionary comment is not related to either the blog post content or the reply-to comment with respect to the topics discussed in the post content. Typical similarity functions such as cosine function [Salton et al. 1974] and Jensen-Shannon divergence [Fuglede and Topsoe 2004] can be used to measure the relatedness between two documents. Here, a document is either a comment or a blog post. Based on our experimental results in a later section, their performance is similar. Cosine similarity between two documents is computed by

$$sim(d_1, d_2) = \frac{\overrightarrow{V}(d_1) \cdot \overrightarrow{V}(d_2)}{|\overrightarrow{V}(d_1)| \cdot |\overrightarrow{V}(d_2)|}, \tag{1}$$

where $\overrightarrow{V}(d_1)$ and $\overrightarrow{V}(d_2)$ are document vectors.[8]

[8]When the topic model is applied, document topic distributions are used as vectors to compute the similarity values.

Jensen-Shannon divergence is a measure of the distance between two probability distributions P and Q. It is based on Kullback-Leibler (KL) divergence [Kullback 2008], but it is symmetric and always a finite value. The Jensen-Shannon divergence of two probability distributions P and Q can be computed by the following function, where $M = \frac{1}{2}(P + Q)$.

$$JSD(P, Q) = \frac{1}{2}(D_{kl}(P, M) + D_{kl}(Q, M)) \tag{2}$$

$$D_{kl}(P, M) = \sum_i P(i) \log \frac{P(i)}{M(i)} \tag{3}$$

However, a simplistic way of utilizing these similarity functions may yield inaccuracies due to the following reasons:

(1) Words that are common to both documents are often replaced by pronouns. Thus, the number of common content words between two related documents is reduced. Coreference resolution [Bengtson and Roth 2008], which ensures that pronouns referring to entities are replaced, is employed to alleviate this problem.
(2) Entities and events mentioned in one document can be different from but closely related to those appearing in another document. Wikipedia[9] provides information about related entities. If a blog post discusses an entity and a comment contains information about a related entity, then proper extraction of information from Wikipedia allows the comment to match the post.
(3) Content words that are different but refer to the same topic are not taken into consideration by typical similarity functions. Topic models such as latent Dirichlet allocation (LDA) [Blei et al. 2003] and hierarchical Dirichlet process (HDP) [Teh et al. 2004] allow different related words to be found that belong to the same topics with high high probability. This also enables similarities to be computed more accurately. Our experimental results in Section 6.3 will show that the topic model–based methods turn out to be very effective.

5.1.1. Coreference Resolution. Coreference resolution groups all mentioned entities in a document into equivalence classes so that all mentions in a class refer to the same entity. By applying coreference resolution, pronouns are mapped into the proper nouns or other noun phrases. If we replace pronouns with their corresponding words or phrases, then the entities become more frequent. For example, a blog post that talks about what Obama has done since he was elected as president only mentions "Obama" once but uses "he" several times. Without coreference resolution, the word "Obama" only occurs once. However, with coreference resolution, "he" will be replaced by "Obama" and the frequency of "Obama" will be increased.

In this work, we use the Illinois coreference package [Bengtson and Roth 2008], which is built on a pairwise classification model. Their idea is to represent mentions in each document by a graph with mentions as nodes. Each mention is first compared to its preceding mentions in the document if it exist, and it is then decided to be linked to the one that returns the highest coreference value. Finally, all connected nodes belong to the same class and refer to the same entity. Here the coreference value indicates the probability of two mentions belonging to the same class and is returned by the pairwise coreference model, which takes mentions' features, such as mention types, string relation, and semantic features, as input. The pairwise coreference model is learned using an averaged perceptron learning algorithm [Freund and Schapire 1999]

[9]http://www.wikipedia.org/.

Table I. Top Terms of an LDA Model

T1	T2	T3	T4
Obama (0.13)	Health (0.80)	Obama (0.20)	War (0.70)
President (0.08)	Care (0.66)	Democrat (0.37)	Iraq (0.96)
Black (0.43)	Tax (0.71)	Party (0.63)	World (0.24)
House (0.33)	Insurance (1.00)	Vote (0.52)	Country (0.22)
Barack (0.47)	Pay (0.75)	People (0.15)	Afghanistan (1.00)

based on a large set of training data. We apply the coreference resolution algorithm to each paragraph separately, as pronouns usually only refer to proper nouns or other noun phrases in the same paragraph.

5.1.2. Extraction from Wikipedia. When a blog post talks about former Chinese president Hu Jintao's visit to the United States, a comment that discusses the foreign policy of China will be considered relevant. However, the blog post does not mention the word "China" and does not share any words with the comment. A similarity function such as cosine, which utilizes words in common, would yield a small value between the post and the comment. Even with coreference resolution, the relationship between "China" and "President Hu Jintao" cannot be detected. Wikipedia comes to help, which offers a vast amount of domain-specific world knowledge. In the preceding example, if we search "President Hu Jintao" in Wikipedia, we will find information that President Hu Jintao is the former president of the People's Republic of China. However, Wikipedia offers much more knowledge than is needed in the analysis of the post or comments. To avoid adding noise, we only pick up anchor texts [Gabrilovich and Markovitch 2007] in the first paragraph from the searched webpage, as this information is believed to be most related.

5.1.3. Latent Dirichlet Allocation. If a similarity function such as the cosine function is applied to two related but different terms, the similarity score will be zero. LDA places different terms, which are related and co-occur frequently, into same topics with high probabilities. Each term can be represented as a vector of topics. Thus, two related terms that share some topics together will have a positive similarity.

Table I lists the top five terms for four different topics (additional topics are not included in the table) in an LDA model, which is built on 600 documents that are related to the query "what Obama has done." In addition, the probability of a word belonging to a topic is listed. From Table I, we find that terms "health" and "insurance" share topic 2, and therefore two comments—one having "health" and the other having "insurance"—can have a positive similarity.

In general, a document-topic distribution can be obtained in the LDA model [Blei et al. 2003; Steyvers and Griffiths 2007; Griffiths and Steyvers 2004] using Gibbs sampling [Heinrich 2004], and it is given by formula (4):

$$\Theta = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} C_{dk}^{DT} + T\alpha}. \tag{4}$$

Here, D and T stand for documents and the number of topics, respectively; C_{dj}^{DT} is the number of occurrences of terms in document d that have been assigned to topic j; and α is a smoothing constant. Based on formula (4), the distribution of a document on a set

of topics can be estimated. Given each document's topic distribution, we can compute the similarity between documents using their topic distribution vectors.

Using Gibbs sampling, a term-topic distribution is also obtained and is given byformula (5):

$$\varphi = \frac{C_{ij}^{WT} + \beta}{\sum_{k=1}^{W} C_{kj}^{WT} + W\beta}. \tag{5}$$

Here, W and T stand for the number of terms and topics, respectively; C_{ij}^{WT} is the number of times that term i has been assigned to topic j; and β is a smoothing constant. This formula allows the similarity between two terms to be computed.

5.1.4. LDA Inference on Test Data. To build an accurate LDA model, a substantial amount of data is required. A blog post and its associated comments usually have limited amount of data. To obtain enough data, we submit the title of the post as a query to search engines and obtain the first 600 documents as preliminary data to build an LDA model. We denote the data as the training data, although no data have been manually labeled in any way, as LDA is an unsupervised model. The post and the associated comments are denoted as test data. Gibbs sampling is still applied to determine the topic assignment for each term occurrence in the test data based on formula (6):

$$\varphi' \propto \begin{cases} \varphi, & \text{if term } i \text{ occurred in the training data,} \\ \frac{C_{ij}^{W'T'} + \beta}{\sum_{k=1}^{W'} C_{kj}^{W'T'} + W'\beta}, & \text{if term } i \text{ only occurs in the test data,} \end{cases} \tag{6}$$

where φ represents the term-topic distribution from the LDA model built on the training dataset; W' and T' stand for the number of terms and topics in the test data, respectively; and $C_{ij}^{W'T'}$ is the number of times that term i has been assigned to topic j in the test dataset. Finally, after the assignment of topic to all term occurrences in the test dataset, the document-topic distribution for each document in the test dataset is obtained by formula (4). In later computation of pairwise similarities based on cosine function or Jensen-Shannon divergence, we use the obtained document-topic distribution in the test data as the document vectors.

5.1.5. Hierarchical Dirichlet Process. When utilizing LDA to learn the topics in a dataset, we need to set the number of preliminary topics. The choice of the number of topics can lead to different results. A model built with too few topics will generally result in very broad topics, whereas a model with too many topics will result in uninterpretable topics. Therefore, a method that could choose the number of topics automatically is desirable. HDP [Teh et al. 2004] tries to extend LDA by using Dirichlet processes to capture the uncertainty regarding the number of topics. The Dirichlet process could be considered as a probability distribution whose domain is also a random distribution.

We apply the Chinese restaurant franchise scheme [Teh et al. 2004] to simulate the HDP, which is a two-level sampling process for our work. Under this setup, each word instance is first assigned to a table (the first level), then a table is assigned to a topic when it is first built (the second level). Tables are local to documents,[10] whereas topics are global across documents. All word instances in a table share the same topic, and different tables could belong to the same topic.

[10]It indicates that documents do not share any tables.

Table II. Notation Used in HDP Posterior Distribution

Notation	Description
t_{ji}	Table assignment for the i^{th} word in the j^{th} document (x_{ji})
$n_{jt.}^{-ji}$	Number of words in the t^{th} table in the j^{th} document except the current word
n_{jtk}	Number of words in the t^{th} table in the j^{th} document belonging to topic k
$n_{..k}$	Number of words being assigned to topic k in the dataset
$m_{.k}$	Number of tables belonging to topic k in the dataset
$m_{..}$	Number of tables in the dataset
$c_k(x_{ji})$	Number of times word x_{ji} being assigned to topic k
V	Number of distinct words in the dataset
K	Number of existing topics in the dataset
$f_t^{-x_{ji}}(x_{ji})$	Conditional probability of assigning word x_{ji} to an existing topic t
$p(x_{ji}\|t^{-ji}, t_{ji} = t^{new}, k)$	Conditional probability of assigning word x_{ji} to a topic

During the sampling process, in each document a word instance could either be assigned to an existing table or a new table.[11] As shown in formula (7), the posterior probability of assigning a word x_{ji} into an existing table t is proportional to the product of the number of existing words in table t and the probability of assigning the word x_{ji} into topic k, which is the topic assignment of the table t; the posterior probability of assigning the word x_{ji} into a new table is proportional to the product of the prior α_0 and the expected value of the probability of assigning the word into a topic (see Equation (9)). When a new table is created, we also need to assign it into a topic, which could either be an existing one or a new one. As shown in formula (10), the posterior probability for assigning the newly built table t into an existing topic k is proportional to the product of the number of tables in topic k and the probability of assigning the word x_{ji} into topic k[12]; the posterior probability for assigning table t into a new topic is proportional to the prior γ. All related notations in the following formula are tabulated in Table II.

$$p(t_{ji} = t|t^{-ji}, k) \propto \begin{cases} n_{jt.}^{-ji} f_t^{-x_{ji}}(x_{ji}), & \text{if t is previously used,} \\ \alpha_0 p(x_{ji}|t^{-ji}, t_{ji} = t^{new}, k), & \text{if } t = t^{new}, \end{cases} \tag{7}$$

where α_0, β and γ are priors,

$$f_t^{-x_{ji}}(x_{ji}) = \frac{c_k(x_{ji}) + \beta}{n_{..k} + V\beta} \tag{8}$$

$$p(x_{ji}|t^{-ji}, t_{ji} = t^{new}, k) = \sum_{k=1}^{K} \frac{m_{.k}}{m_{..} + \gamma} f_t^{-x_{ji}}(x_{ji}) + \frac{\gamma}{m_{..} + \gamma} \frac{1}{V} \tag{9}$$

$$p(k_{jt^{new}} = k|t, k^{-jt^{new}}) \propto \begin{cases} m_{.k} f_t^{-x_{ji}}(x_{ji}), & \text{if } k \text{ is previously used;} \\ \frac{\gamma}{V}, & \text{if } k = k^{new}. \end{cases} \tag{10}$$

Finally, after the sampling process, all word instances in each document are grouped into several tables, whereas some tables belong to the same topic. To get the topic distribution of each document, we should not simply count the number of words being

[11]In the first iteration of the sampling process, the first word instance of each document is always assigned to a new table, because initially there is no table for each document.

[12]This new table is created only because the word x_{ji} is assigned into a new table. At this moment, the table only contains the word x_{ji}.

ALGORITHM 1: Rank comments in descending order of being diversionary

Constants t_1, t_2, t_3, t_4, where $t_1 \leq t_3$, and $t_2 \leq t_4$
for *each comment* **do**
  C_1 = the similarity between the comment and the post;
  C_2 = the similarity between the comment and its reply-to comment;
  if *its level* == 0 *and* $C_1 > C_2$ *and* $C_1 \geq t_3$ **then**
    $C_2 = C_1$;
  end
  if $C_1 < t_1$ *and* $C_2 < t_2$ **then**
    Put the comment into potential diversionary list (PDL);
  else if $C_1 > t_3$ *or* $C_2 > t_4$ **then**
    Put the comment into potential nondiversionary list (PNDL);
  else
    Put the comment into the intermediate list (IL);
  end
end
Sort comments in PDL in ascending order of sum(C_1, C_2);
Sort comments in IL in ascending order of max($C_1 - t_1$, $C_2 - t_2$);
Sort comments in PNDL in ascending order of max($C_1 - t_3$, $C_2 - t_4$);
Output comments in PDL followed by comments in IL, followed by comments in PNDL.

assigned to each topic under each document, as this would ignore the diversity between different tables under the same topic in a document. Regarding this issue, we consider a document as a multidimensional space spanned by the tables in the document. Each table represents one dimension, and the topics are vectors in the space. Then for each topic, its projection to a dimension is the number of word instances in that dimension (or table) being assigned to it. Therefore, the magnitude of a vector (i.e., the topic k) is $\sqrt{\sum_t n_{jtk}^2}$ and then a document j's topic distribution is computed by formula (11):

$$\Theta_j = \frac{\sqrt{\sum_t n_{jtk}^2}}{\sum_{k=1}^{K} \sqrt{\sum_t n_{jtk}^2}}. \tag{11}$$

After building an HDP model on the training data, we also need to infer the topic distribution for each document in the test data. The preceding sampling process is still applied, but the starting number of topics for the test data is the number of existing topics in the training data, and $n_{\cdot k}$, m_k, and $m_{\cdot}$ from the training data are used for sampling process of the test data. The topic distribution for each document in the test data is computed by formula (11).

5.1.6. Rank Comments in Descending Order of Being Diversionary. According to the property that a diversionary comment is unrelated to both the blog post content and its reply-to comment with respect to topics in the post, if a comment has small similarities to both the blog post and the reply-to comment, then there is a high probability of it being diversionary. As a consequence, we set two thresholds t_1 and t_2 such that if a comment's similarity with the blog post (C_1) is less than t_1 and its similarity with the reply-to comment (C_2) is less than t_2, then it is placed into a list called the *potential diversionary list* (PDL). Within this list, the smaller the sum of the two similarities, the more likely it is to be diversionary. Thus, comments in this list are sorted in ascending order of sum(C_1, C_2), with the first one most likely to be a diversion.

In contrast, if a comment has a big enough similarity either to the blog post or to its reply-to comment, it is very unlikely to be diversionary.[13] As a result, we set two thresholds t_3 and t_4 such that if the similarity of a comment to the post is higher than t_3, or its similarity to its reply-to comment is higher than t_4,[14] then it is placed into a list called the *potential nondiversionary list* (PNDL). The more the similarity between the comment and the post (C_1) differs from t_3, or the more the similarity between the comment and its reply-to comment (C_2) differs from t_4, the less likely the comment is diversionary. Thus, comments within the PNDL are sorted in ascending order of $\text{Max}(C_1 - t_3, C_2 - t_4)$.

Comments that belong to neither of the preceding two lists are placed into an *intermediate list* (IL). Comments in this list do not have a high probability of being diversionary relative to those in PDL; they also do not have a high probability of being nondiversionary compared to those in PNDL. Thus, comments in PDL are placed ahead of comments in IL, which are ahead of comments in PNDL. Within IL, the more the similarity between the comment and the blog post (C_1) differs from t_1, or the more the similarity between the comment and its reply-to comment (C_2) differs from t_2, the less likely the comment is diversionary. Therefore, they are sorted in ascending order of $\text{Max}(C_1 - t_1, C_2 - t_2)$. We will discuss how to set the threshold values and study their sensitivity in Section 6.3.

Based on the preceding analysis, we use Algorithm 1 to rank comments.

5.2. Pipeline of the Proposed Method

Our proposed method combines the techniques discussed earlier to identify diversionary comments. Figure 3 provides a pipeline of the method. Each step in the procedure is described as follows:

(1) Submit each blog post title as a query to two search engines (Bing and Yahoo), and retrieve all of the returned Web pages. Among all retrieved Web pages, we extract contents from them—up to 600 Web pages as the training corpus—and consider them related to the post content. The test corpus consists of each post and the associated comments.
(2) Apply coreference resolution on each paragraph of each document in the training corpus and the test corpus separately, and replace pronouns with their corresponding proper nouns or phrases. This is useful for building an accurate topic model (LDA or HDP) in later steps.
(3) Identify proper nouns in the test data based on the Stanford POS tagger [Toutanova et al. 2003], and a dictionary indexed by these proper nouns is built based on Wikipedia in the following way. For each of the proper nouns, search it through Wikipedia; if an unambiguous page is returned, terms in the anchor texts in the first paragraph of the page are added into the dictionary as the related terms of the proper noun. Then for each document in the training and test corpus, if it contains a proper noun in the dictionary, we add the corresponding related terms into the document.
(4) Build an LDA (or HDP) model based on the training data first, then use it to infer the document-topic distribution for documents in the test data.

[13]It is possible that a comment has a high similarity to its reply-to comment and that its reply-to comment is a diversion, which causes a problem in our approach. However, such cases are rare. In our dataset, there are only 0.76% such comments. We will deal with this problem in future work. Our current approach works quite accurately, as the experimental results show.

[14]As mentioned in Section 1, the criteria for "close relatedness" (in the pattern of similarity score) between a comment and its reply-to comment is higher than that between a comment and the post content—t_4 is set bigger than t_3.

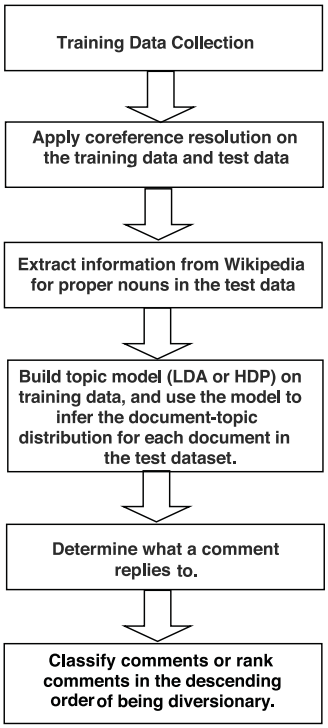

Fig. 3. Pipeline of the proposed method.

(5) According to the rules described in Section 4.1, measure similarities between each comment and the blog post, and similarities between each comment and its potential reply-to comments in the test corpus, and then decide what a comment replies to. Similarity here is measured by computing the cosine value or Jensen-Shannon divergence between two documents' topic distributions.
(6) Rank comments based on Algorithm 1, or classify the comments into diversion or nondiversion by using the similarity between each comment and the post, and the similarity between each comment and its reply-to comment, as features. The linear SVM algorithm is applied.

6. EVALUATION

For our experiments, we collected data from Digg and Reddit. The dataset from Digg contains 20 blogs and 2,109 associated comments dating from October 2010 to February 2011.[15] The corpus was annotated by five annotators, all of whom were graduate students. Each comment was assigned to be a nondiversionary comment or one of the five types of diversions. When the annotators assessed whether the topic of a comment was different from that of the blog post, they used the criteria of whether the topic of the post was mentioned in the comment. Among all annotators, one of them completed the annotations of the comments of all 20 posts, two completed the annotations of the comments for the first 10 posts, and another two completed the annotations of the comments for the remaining 10 posts. All annotators resolved the disagreement in the annotations together. We consider the final annotation to be the gold standard.

[15]We began this work around that time, so the data was randomly collected then.

Table III. κ Agreement and Agreement Score Percentage for the Diggs Dataset

Annotator Pair	κ	%
(A1,A2)	0.61	0.82
(A1,A3)	0.63	0.85
(A2,A3)	0.67	0.86
(A4,A5)	0.67	0.85
(A4,A3)	0.65	0.85
(A5,A3)	0.57	0.80
Average	0.63	0.84

Table IV. κ Agreement and Agreement Score Percentage for the Reddit Dataset

Annotator Pair	κ	%
(A6,A7)	0.63	0.81
(A8,A9)	0.65	0.84
Average	0.64	0.83

The dataset from Reddit contains 20 blogs and 2,070 associated comments, which were collected around October 2013.[16] The corpus was annotated in the same way, except there were only four annotators. Among them, two annotators completed the annotation of comments in the first 10 posts, and another two completed the annotations of the remaining comments.

6.1. Interannotator Agreement for Diversionary Comments Annotation

This section reports on an agreement study that was conducted to measure the reliability of the various annotations. We use Cohen's kappa coefficient [Cohen 1960] to measure the agreement of each pair of annotators, which is believed to be more appropriate than the simple agreement calculation percentage, as it takes into account the chance agreement between annotators. It measures the agreement between two annotators, each of whom classifies N items into C mutually exclusive categories. In its most general form, κ is defined to be

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}, \tag{12}$$

where $Pr(a)$ is the observed agreement among annotators and $Pr(e)$ is a measure of the agreement that can be expected by chance. Cohen's κ ranges between 0 and 1, with $\kappa = 1$ indicating perfect agreement and $\kappa = 0$ indicating agreement that is not better than chance. We list the pairwise κ agreement values for each pairwise annotators in Tables III and IV. For comparison, the absolute agreement score percentages are also given. In interpreting κ, Landis and Koch [1977] suggest that values greater than 0.61 indicate substantial strength of agreement, and therefore we believe that our annotation results are enough for at least tentative conclusions.

6.2. Diversionary Comments Distribution

In this section, we report diversionary comments distribution variation. Based on the gold standard, there are 834 diversionary comments in the Digg dataset (accounting for 39.5% of all Digg comments) and 449 diversionary comments in the Reddit dataset (accounting for 21.7%). Figure 4 provides the diversionary comments distribution across different blog posts; the first 20 points represent distributions of posts from Digg, whereas the last 20 represent posts from Reddit. We observe that most blog posts from Digg contain 35% to 45% of diversionary comments, whereas most Reddit posts contain around 25% diversionary comments. Figure 5 gives the distribution of different types of diversionary comments. It shows that among all diversionary comments, type 1 is the most significant one, whereas type 5 has the lowest percentages in both Digg and

[16]It was the time we started revising this work.

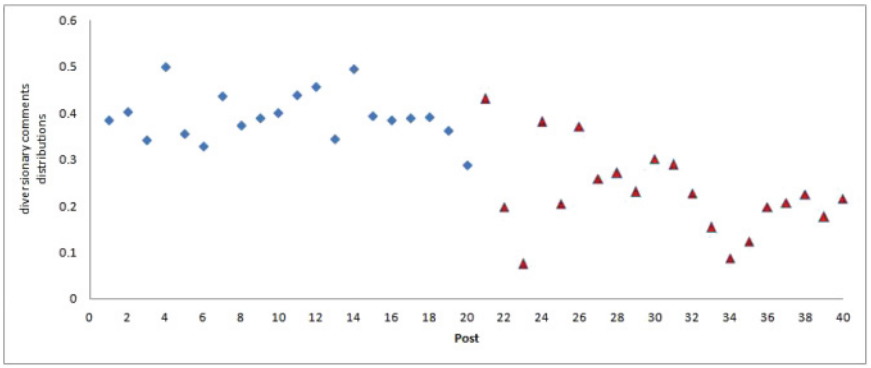

Fig. 4. Diversionary comments distribution in each blog post.

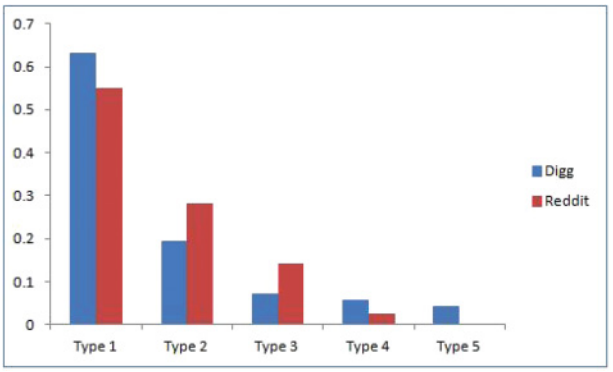

Fig. 5. Different types of diversionary comments distribution.

Reddit datasets (in the Reddit dataset, there are no type 5 diversions), which also indicates that diversionary comments studied in this work are not commercially oriented but focus on those diverting to other topics.

6.3. Experimental Results

As we proposed in Section 4, our method consists of several techniques. To test the necessity of combining them, we performed experiments by comparing our final method to baseline methods that only apply one technique or combine fewer techniques. We first evaluate our approach of ranking comments in descending order of being diversionary. The effectiveness of each method is measured by MAP [Manning et al. 2008].

To keep consistency among all methods being compared, we set parameters t_1, t_2, t_3, and t_4 using fixed percentiles, which are required in the ranking algorithm as presented in Algorithm 1. Let us explain the setting of these parameter values with an example: if t_1 equals to 10%, then the similarities between the comments and the blog post are sorted in ascending order and the similarity at the top 10% is assigned to t_1. In the following section, we set t_1 equal to 10%, t_2 equal to 20%, t_3 equal to 50%, and t_4 equal to 90%.[17]

6.3.1. Results by Adding Techniques Step by Step. We first compare the following methods: cosine similarity with term frequency, cosine similarity with coreference resolution, cosine similarity with extraction from Wikipedia, and cosine similarity with both

[17]Their values are tuned based on a subset (25%) of the dataset.

Table V. MAP for Cosine Similarity with Document Vectors Built by Term Frequency

Posts Methods Results	Term Frequency (%)	With Coreference Resolution (%)	With Wikipedia Extraction (%)	With Coreference and Wikipedia (%)
Digg-1	52.9	55.2	57.1	57.3
Digg-2	75.6	74.4	72.1	70.7
Digg-3	62.1	65.9	67.9	68.2
Digg-4	85.8	83.0	92.5	90.3
Digg-5	79.5	79.0	79.6	79.0
Digg-6	54.9	56.8	58.9	60.7
Digg-7	70.9	73.0	79.3	79.7
Digg-8	75.9	75.6	75.7	75.1
Digg-9	51.2	51.9	49.9	52.8
Digg-10	70.2	73.0	74.6	76.5
Digg-11	79.1	78.5	81.9	81.1
Digg-12	81.8	81.6	81.9	82.2
Digg-13	67.1	66.5	68.5	67.9
Digg-14	83.7	84.6	84.0	83.8
Digg-15	79.5	81.3	80.5	81.8
Digg-16	61.2	60.7	60.4	60.4
Digg-17	76.3	77.0	77.5	78.4
Digg-18	83.0	83.0	82.9	82.2
Digg-19	54.6	54.3	54.4	57.1
Digg-20	77.0	77.0	80.9	80.9
Reddit-1	71.0	71.7	70.6	70.5
Reddit-2	56.0	56.0	47.3	47.2
Reddit-3	17.0	17.6	17.6	18.3
Reddit-4	60.6	62.2	61.3	62.2
Reddit-5	48.6	43.6	50.9	45.8
Reddit-6	66.0	65.6	64.2	63.9
Reddit-7	52.6	52.6	52.6	52.6
Reddit-8	64.0	64.6	63.8	64.4
Reddit-9	53.6	53.5	53.6	53.5
Reddit-10	72.5	72.5	72.8	72.6
Reddit-11	63.6	60.5	63.6	60.5
Reddit-12	55.3	55.2	55.3	55.2
Reddit-13	40.6	40.3	43.2	43.2
Reddit-14	34.8	35.2	34.8	35.2
Reddit-15	41.3	41.1	41.5	41.5
Reddit-16	43.5	46.0	43.5	46.0
Reddit-17	41.7	42.2	41.1	41.9
Reddit-18	48.1	51.1	48.1	51.0
Reddit-19	57.7	58.5	58.8	59.6
Reddit-20	57.7	60.5	58.6	62.4
MAP	**61.1**	**61.5**	**61.9**	**62.3**

coreference resolution and extraction from Wikipedia in Table V. All of these methods represent comments and the post by building vectors based on term frequencies. From Table V, we observe that cosine similarity with term frequency has the lowest MAP value, whereas cosine similarity with both coreference resolution and extraction from Wikipedia performs the best. Yet even the best result is far from being acceptable. The reasons for these poor results are obvious. Cosine similarity by term frequency

Table VI. MAP for Cosine with Document Vectors Built by LDA on Test Data

Posts Methods Results	LDA on Test Data (%)	LDA with Coreference Resolution (%)	LDA with Wikipedia Extraction (%)	LDA with Coreference and Wikipedia (%)
Digg-1	57.4	47.9	53.4	66.5
Digg-2	68.5	54.3	52.4	57.8
Digg-3	53.9	51.4	51.9	61.7
Digg-4	77.9	74.9	84.8	93.3
Digg-5	62.9	66.7	58.6	72.8
Digg-6	52.9	55.9	59.5	55.9
Digg-7	59.1	61.6	67.0	67.0
Digg-8	71.3	67.5	72.1	72.1
Digg-9	58.1	42.6	50.9	50.9
Digg-10	67.1	66.2	73.2	73.2
Digg-11	51.0	53.0	67.8	67.8
Digg-12	64.1	67.6	69.2	75.1
Digg-13	42.8	58.1	53.0	53.7
Digg-14	75.2	70.5	80.5	80.5
Digg-15	39.6	49.8	57.7	57.7
Digg-16	57.6	56.3	57.2	57.2
Digg-17	54.0	60.3	58.7	58.7
Digg-18	52.0	54.9	64.6	68.2
Digg-19	40.0	47.1	47.1	47.1
Digg-20	42.8	52.9	43.5	52.9
Reddit-1	57.3	51.8	55.2	57.9
Reddit-2	31.1	30.7	29.6	30.1
Reddit-3	22.3	11.3	24.4	15.9
Reddit-4	56.3	51.1	53.5	48.2
Reddit-5	37.4	34.9	28.2	34.4
Reddit-6	68.5	57.9	61.0	50.1
Reddit-7	38.6	35.8	35.6	40.3
Reddit-8	25.1	27.6	29.7	34.6
Reddit-9	35.0	41.5	35.6	43.8
Reddit-10	48.8	39.5	46.3	39.3
Reddit-11	30.2	29.0	31.5	33.7
Reddit-12	35.9	57.0	35.9	57.0
Reddit-13	17.0	37.0	26.5	31.7
Reddit-14	12.5	14.9	12.5	14.9
Reddit-15	14.6	23.3	20.8	12.7
Reddit-16	31.3	32.5	31.3	32.5
Reddit-17	35.9	43.6	29.9	26.2
Reddit-18	29.3	51.4	25.2	27.8
Reddit-19	41.4	16.9	37.1	38.9
Reddit-20	26.5	27.8	31.2	30.5
MAP	**46.1**	**46.9**	**47.6**	**49.8**

is incapable of matching a document with another document if they have related but different terms. This mismatch can be alleviated to some extent by coreference resolution, by extracting related information from Wikipedia, and a combination of the two techniques. However, many unrelated terms remain unmatched.

When LDA is applied, the number of topics is set to 10, α to 0.1, and β to 0.01. In Table VI, when the LDA model is built simply on the test data, we represent comments

Table VII. MAP for Cosine with Document Vectors Built by LDA Inference on Test Data

Posts Methods Results	LDA Inference on the Test Data (%)	LDA with Coreference Resolution (%)	LDA with Wikipedia Extraction (%)	LDA with Coreference and Wikipedia (%)
Digg-1	66.5	65.1	69.6	86.8
Digg-2	68.3	76.0	83.0	97.8
Digg-3	75.9	84.7	84.0	89.8
Digg-4	83.3	76.4	88.8	96.5
Digg-5	84.5	82.6	83.6	84.1
Digg-6	62.3	57.1	66.3	85.1
Digg-7	60.8	59.3	82.2	95.3
Digg-8	75.2	77.4	89.8	90.4
Digg-9	65.0	72.7	68.9	93.6
Digg-10	82.6	86.8	74.3	96.7
Digg-11	70.3	68.5	83.6	94.9
Digg-12	85.5	85.0	88.5	96.8
Digg-13	79.2	76.2	82.7	97.3
Digg-14	89.2	89.7	91.4	92.7
Digg-15	87.7	89.6	89.5	95.1
Digg-16	80.4	79.2	78.5	97.6
Digg-17	85.6	79.6	76.3	95.6
Digg-18	70.6	80.8	86.4	96.2
Digg-19	64.9	73.9	66.1	88.6
Digg-20	70.0	73.0	82.0	82.0
Reddit-1	72.3	68.5	80.7	89.5
Reddit-2	90.5	92.3	93.1	98.5
Reddit-3	66.7	70.9	78.0	85.2
Reddit-4	74.3	79.2	79.4	93.8
Reddit-5	75.9	80.0	73.2	90.4
Reddit-6	80.7	77.5	83.6	90.2
Reddit-7	81.1	84.8	83.7	88.5
Reddit-8	77.2	92.1	70.0	91.8
Reddit-9	88.5	82.7	92.4	91.8
Reddit-10	78.4	80.3	76.6	88.6
Reddit-11	73.6	64.1	75.5	93.5
Reddit-12	76.9	81.2	80.8	89.4
Reddit-13	54.1	46.7	65.4	92.3
Reddit-14	74.3	80.8	76.9	87.1
Reddit-15	83.5	75.5	82.4	98.6
Reddit-16	78.2	72.6	67.6	90.1
Reddit-17	89.1	89.2	80.9	91.6
Reddit-18	84.1	77.8	82.4	90.7
Reddit-19	83.6	75.4	72.2	91.3
Reddit-20	87.9	83.2	84.6	91.1
MAP	**77.0**	**77.2**	**79.9**	**91.9**

and the post by their topic distributions. However, the results are also poor. When coreference resolution, extraction from Wikipedia, or both are combined with LDA, better results are more often obtained. However, even the best result in this table has a MAP value of only 49.8%. The reason for such a poor result is that the amount of test data is too small for LDA to learn reasonable topics.

In Table VII, the LDA inference is applied to the post and comments in the test data. We rank comments in the test dataset based on the cosine similarities of their topic

distributions. If the entries in the second column of Table VII are compared against those in the second column of Tables V and VI, there is a major improvement, implying that this LDA inference method does find related terms across all comments and the post. When coreference resolution and extraction from Wikipedia are individually added in, there are notable improvements. The largest and most dramatic improvement comes when LDA and the two techniques are combined, yielding 91.9% MAP.

In Table VIII, we report the results when the cosine similarity function is replaced with the Jenson-Shannon divergence function. The parameter values remain unchanged. The results turn out to be close to those in Table VII, where the cosine similarity is applied.

When HDP is applied, α_0, β, and γ are all set to 1.0 [Teh et al. 2004], and there is no need to preset the number of topics. In Table IX, we report the result of HDP built on the test data directly with coreference resolution and extraction from Wikipedia. The number of topics learned from HDP is listed as well. The performance turns out to be close to that of LDA built on the test data directly with coreference resolution and extraction from Wikipedia, as listed in the fifth column of Table VI.

In Table X, we report the performance of HDP inference on the test data with coreference resolution and extraction from Wikipedia. The number of topics learned from HDP is listed as well. The result is similar to that of LDA inference on the test data with coreference resolution and extraction from Wikipedia, as listed in the fifth column of Table VII. Therefore, HDP is proved to achieve comparable performance with LDA without the need to specify the number of topics.

6.3.2. Diversionary Comments Classifier. In this section, we evaluate our approach when the classification algorithm is applied. The similarity between each comment and the post, and the similarity between each comment and its reply-to comment, are taken as features. Then the linear SVM algorithm [Bishop 2007] is applied to build the classifier. To show the effectiveness of our method, we compare our method with the method Ott et al. [2011] used to detect deceptive opinion spam. They studied hotel reviews and also used the linear SVM classifier, but their features contain unigrams and bigrams in the reviews, and features extracted from LIWC software. LIWC counts and groups the number of instances of nearly 4,500 keywords into 80 dimensions, such as total word count, words per sentence, percentage of words captured by a psychological dictionary, percentage of words in the text that are pronouns, and articles. Weighted precision, weighted recall, and weighted F-measure are calculated based on 10-fold cross-validation. Weighted F-measure is the weighted sum of two F-measures, one with respect to diversionary comments and the other with respect to nondiversionary comments, each weighted according to the number of instances with that particular class label. Weighted precision and weighted recall are calculated in a similar way. The results are reported in Table XI, in which it is shown that our method obtains 84.9% as the average F-measure, whereas their method only achieves 59.0% F-measure on average. In addition, we observe that the F-measure across different posts obtained from their method are much more diverse, whereas our method provides a stable F-measure across different blog posts.

We also studied the performance of the classifier that takes the document-topic distribution as features. For each comment, without considering what it replies to, we take its probabilities of being assigned to each topic as inputs to the linear SVM classifier. The results are reported in Table XII. This way of building features returns an average F-measure of 70.7%, which is also better than the method that Ott et al. use to detect deceptive opinion spam but worse than our proposed method, which indicates the high effectiveness of the heuristic rules that we use to find what a comment replies to.

Table VIII. MAP for Jenson-Shannon Divergence Function with Document Vectors Built by LDA Inference on the Test Data

Posts Methods Results	LDA Inference on the Test Data (%)	LDA with Coreference Resolution (%)	LDA with Wikipedia Extraction (%)	LDA with Coreference and Wikipedia (%)
Digg-1	64.3	67.1	67.1	77.6
Digg-2	70.1	78.5	81.9	93.8
Digg-3	68.2	72.9	83.5	86.6
Digg-4	83.2	80.0	85.2	93.8
Digg-5	82.8	80.8	87.7	88.7
Digg-6	60.0	52.4	59.9	66.8
Digg-7	58.5	59.5	78.7	83.9
Digg-8	77.0	80.9	90.7	91.0
Digg-9	63.6	68.2	70.8	90.4
Digg-10	82.8	87.8	77.2	92.0
Digg-11	67.4	68.4	82.5	90.7
Digg-12	86.3	88.1	88.9	96.4
Digg-13	78.8	76.1	71.7	82.2
Digg-14	93.6	91.9	93.7	93.9
Digg-15	82.6	87.2	83.0	93.9
Digg-16	77.3	74.8	77.5	93.2
Digg-17	85.2	81.2	78.1	89.4
Digg-18	72.5	77.7	80.7	90.5
Digg-19	71.1	70.3	60.1	94.5
Digg-20	72.7	75.7	82.7	92.7
Reddit-1	68.7	63.2	73.5	84.8
Reddit-2	69.4	90.9	90.4	92.0
Reddit-3	57.4	70.9	78.0	87.7
Reddit-4	72.2	76.6	78.0	89.5
Reddit-5	77.0	80.5	73.4	90.4
Reddit-6	78.6	77.9	82.7	86.6
Reddit-7	72.8	85.2	80.9	89.8
Reddit-8	72.7	85.2	70.0	81.3
Reddit-9	89.2	82.2	79.4	96.0
Reddit-10	73.6	80.3	76.6	89.4
Reddit-11	72.3	66.4	60.8	93.5
Reddit-12	76.9	62.1	80.8	89.4
Reddit-13	54.1	46.7	65.4	92.3
Reddit-14	74.9	68.7	75.8	87.1
Reddit-15	78.1	72.8	78.7	92.9
Reddit-16	77.8	72.6	67.6	89.1
Reddit-17	86.0	83.1	75.3	91.6
Reddit-18	66.3	77.8	82.4	79.2
Reddit-19	62.8	68.1	72.2	91.3
Reddit-20	90.8	77.9	84.6	80.3
MAP	**74.2**	**75.2**	**77.8**	**88.9**

6.3.3. Accuracy of Finding What a Comment Replies to. When applying our method to compute pairwise similarities, the heuristic rules given in Section 4.1 provide 98.2% precision and 100% recall for locating the reply-to comment of a comment. In addition, when setting the threshold t (see Section 4.1) equal to 50%, the precision is 100% and the recall is 81.8% for recognizing comments replying to the post directly.

Table IX. MAP for Cosine with Document Vectors Built by HDP on Test Data

Posts Methods Results	HDP on the Test Data (%)	Topics (#)
Digg-1	49.4	12
Digg-2	64.7	10
Digg-3	61.1	11
Digg-4	92.1	14
Digg-5	56.9	12
Digg-6	56.1	9
Digg-7	67.8	11
Digg-8	66.8	11
Digg-9	56.7	14
Digg-10	64.2	8
Digg-11	78.3	12
Digg-12	70.8	14
Digg-13	60.3	11
Digg-14	65.5	11
Digg-15	61.0	13
Digg-16	61.3	11
Digg-17	68.9	10
Digg-18	70.8	13
Digg-19	62.9	15
Digg-20	61.7	13
Reddit-1	48.2	9
Reddit-2	40.0	13
Reddit-3	24.4	11
Reddit-4	46.7	12
Reddit-5	21.8	12
Reddit-6	32.8	9
Reddit-7	26.2	12
Reddit-8	41.9	11
Reddit-9	33.7	8
Reddit-10	36.8	12
Reddit-11	32.6	10
Reddit-12	47.4	13
Reddit-13	20.6	15
Reddit-14	21.0	10
Reddit-15	35.8	10
Reddit-16	17.2	10
Reddit-17	19.5	7
Reddit-18	32.3	10
Reddit-19	30.9	13
Reddit-20	37.5	9
MAP	**48.6**	

6.3.4. Sensitivity Analysis. To test the stability of our method, we compared its effectiveness by setting different parameter values. We first tested its sensitivity by setting different numbers of topics while keeping other parameter values unchanged when LDA is used to learn the topic distributions in the comments and the post. In Table XIII, the number of topics is set to 6, 8, 12, 20, and 30. Similar MAPs are obtained when the number of topics is 8 and 12, whereas the other numbers provide lower

Table X. MAP for Cosine with Document Vectors Built by HDP Inference on Test Data

Posts Methods Results	HDP Inference (%)	Topics (#)
Digg-1	93.5	9
Digg-2	93.0	8
Digg-3	92.5	9
Digg-4	91.6	8
Digg-5	94.6	10
Digg-6	92.4	8
Digg-7	91.5	9
Digg-8	91.6	10
Digg-9	91.6	9
Digg-10	92.3	10
Digg-11	91.6	9
Digg-12	91.7	9
Digg-13	88.5	12
Digg-14	89.1	13
Digg-15	90.7	10
Digg-16	91.6	12
Digg-17	88.1	10
Digg-18	94.2	6
Digg-19	89.0	10
Digg-20	91.2	8
Reddit-1	85.7	14
Reddit-2	90.2	11
Reddit-3	90.6	10
Reddit-4	88.1	8
Reddit-5	87.7	15
Reddit-6	88.1	9
Reddit-7	90.8	15
Reddit-8	93.0	22
Reddit-9	88.5	20
Reddit-10	91.1	11
Reddit-11	91.5	11
Reddit-12	85.8	8
Reddit-13	87.4	14
Reddit-14	87.6	10
Reddit-15	91.5	17
Reddit-16	90.9	10
Reddit-17	89.6	9
Reddit-18	90.2	10
Reddit-19	90.0	7
Reddit-20	92.4	14
MAP	**90.5**	

performance. Thus, our method with LDA is believed to be stable when number of topics is in a reasonable range (8 to 12).[18]

[18]Although HDP could be utilized to get the number of topics automatically, it is much more time consuming than LDA; therefore, learning the sensitivity of LDA to the number of topics is meaningful.

Table XI. Comparison between the LIWC+Unigram+Bigram Method and the Classifier With Our Approach

Linear SVM Classification Results	LIWC+Unigram+ Bigram			Comment Similarities as Features		
	Precision	*Recall*	*F-Measure*	*Precision*	*Recall*	*F-Measure*
Digg-1	41.4	28.6	33.8	82.4	81.7	81.8
Digg-2	55.0	52.4	53.7	89.6	89.4	89.4
Digg-3	66.7	5.4	10.0	80.4	78.7	77.8
Digg-4	62.0	98.0	76.0	81.1	81.0	81.0
Digg-5	75.0	28.1	40.9	90.0	90.0	89.9
Digg-6	36.4	12.1	18.2	84.4	84.0	83.1
Digg-7	54.2	45.1	49.2	93.3	93.2	93.2
Digg-8	93.3	41.2	57.1	89.0	87.9	87.4
Digg-9	60.6	51.3	55.6	79.5	79.0	79.0
Digg-10	76.5	31.7	44.8	80.9	79.4	79.2
Digg-11	60.8	70.5	65.3	77.9	77.0	77.1
Digg-12	57.5	100	73.0	80.5	80.5	80.5
Digg-13	40.0	11.1	17.4	81.7	80.8	80.8
Digg-14	63.1	98.2	76.8	79.8	78.9	78.8
Digg-15	57.8	90.2	70.5	82.8	82.7	82.6
Digg-16	33.3	10.5	16.0	85.5	85.1	85.1
Digg-17	59.7	87.2	70.8	83.8	83.0	82.2
Digg-18	54.7	87.5	67.3	83.1	82.4	82.2
Digg-19	66.7	6.1	11.1	84.6	84.6	84.6
Digg-20	50.0	15.2	23.3	87.7	87.7	87.7
Reddit-1	63.9	63.8	61.3	79.6	79.3	79.4
Reddit-2	83.4	84.0	80.4	92.2	91.4	90.4
Reddit-3	85.5	92.5	88.9	85.5	92.5	88.9
Reddit-4	66.2	67.3	65.5	84.5	82.7	81.7
Reddit-5	63.3	79.6	70.5	85.3	86.0	85.6
Reddit-6	49.5	58.1	50.5	78.9	79.0	78.4
Reddit-7	84.2	80.0	74.9	82.6	81.7	82.0
Reddit-8	52.9	72.0	61.0	86.2	86.4	86.3
Reddit-9	58.9	75.4	66.1	92.3	92.3	92.0
Reddit-10	82.2	76.2	70.2	80.5	81.0	80.6
Reddit-11	50.2	69.7	58.4	80.0	80.3	80.1
Reddit-12	85.7	82.5	77.9	88.3	88.6	88.3
Reddit-13	71.4	83.7	77.1	71.5	84.6	77.5
Reddit-14	83.2	91.2	87.1	94.2	94.3	94.3
Reddit-15	76.8	87.6	81.9	95.7	95.5	95.1
Reddit-16	84.6	81.0	73.3	88.0	86.0	82.7
Reddit-17	62.8	79.2	70.1	93.3	93.4	93.2
Reddit-18	60.0	76.8	67.4	92.1	92.0	92.1
Reddit-19	80.5	83.1	77.3	93.2	93.1	93.1
Reddit-20	61.5	78.4	69.0	90.8	90.5	89.7
Average Performance	**64.3**	**63.3**	**59.0**	**85.3**	**85.5**	**84.9**

Second, we tested the method's stability by setting different values for ranking algorithm parameters.[19] To make the comparison simple, we set t_1 and t_2 to be the same percentile and t_4 to be the percentage of t_3 plus 10%, and report the results in Table XIV.

[19]The result listed here is obtained by using LDA inference to get the documents' topic distributions. HDP inference gets similar results.

Table XII. Classifier with Comments' Topic Distribution as Features

Linear SVM Classification Results	Topic Distributions as Features		
	Precision	*Recall*	*F-Measure*
Digg-1	56.4	56.4	56.4
Digg-2	56.5	56.2	56.3
Digg-3	61.7	62.4	62.0
Digg-4	55.4	55.4	55.4
Digg-5	62.5	63.7	62.9
Digg-6	64.9	66.3	65.4
Digg-7	56.7	57.1	56.8
Digg-8	69.8	70.7	69.1
Digg-9	53.6	54.5	53.9
Digg-10	57.2	56.3	56.6
Digg-11	54.5	54.5	54.5
Digg-12	68.9	68.9	68.9
Digg-13	60.4	61.9	61.0
Digg-14	78.3	78.2	78.1
Digg-15	69.1	69.5	69.2
Digg-16	50.5	51.0	50.7
Digg-17	65.1	64.4	64.6
Digg-18	50.3	50.5	50.4
Digg-19	67.6	68.5	67.8
Digg-20	63.4	67.8	64.4
Reddit-1	71.6	70.7	69.2
Reddit-2	90.2	88.9	87.1
Reddit-3	85.5	92.5	88.9
Reddit-4	69.6	70.0	67.9
Reddit-5	77.4	80.6	75.8
Reddit-6	71.9	72.6	71.7
Reddit-7	85.5	85.8	85.0
Reddit-8	85.6	85.6	84.5
Reddit-9	78.8	80.0	79.2
Reddit-10	71.0	73.0	71.0
Reddit-11	58.7	67.1	60.6
Reddit-12	83.2	84.2	83.1
Reddit-13	71.5	84.6	77.5
Reddit-14	85.7	90.2	87.3
Reddit-15	97.8	97.8	97.7
Reddit-16	85.9	86.8	86.0
Reddit-17	87.1	87.7	86.8
Reddit-18	82.7	83.3	80.8
Reddit-19	80.6	83.1	81.1
Reddit-20	82.5	83.6	81.3
Average Performance	**70.6**	**72.1**	**70.7**

Here, t_1 and t_2 are set in the range from 0.1 to 0.45, whereas t_3 changes from 0.2 to 0.55 and t_4 changes from 0.3 to 0.65. The MAPs based on cosine function are provided. We find that with such wide ranges of threshold values, there is very little change in the effectiveness of identifying diversionary comments. Therefore, we conclude that the method is stable with reasonable threshold values.

Table XIII. MAP for LDA Inference on the Test Data with Different Numbers of Topics

Posts Results	$T = 6$ (%)	$T = 8$ (%)	$T = 12$ (%)	$T = 20$ (%)	$T = 30$ (%)
Digg-1	62.1	83.8	83.8	68.4	73.4
Digg-2	83.6	95.3	89.2	76.2	69.3
Digg-3	65.0	81.7	88.7	76.2	72.7
Digg-4	81.2	87.9	88.5	83.7	78.9
Digg-5	80.1	86.0	90.9	77.8	79.2
Digg-6	51.6	85.0	91.5	55.7	44.3
Digg-7	74.2	81.7	85.6	77.9	72.9
Digg-8	69.0	89.5	85.9	76.8	70.6
Digg-9	55.7	87.3	91.3	61.6	65.6
Digg-10	77.4	90.3	94.5	80.1	76.8
Digg-11	64.2	82.5	85.3	68.5	66.0
Digg-12	83.2	92.3	88.0	83.7	78.6
Digg-13	69.1	81.1	88.2	75.2	73.7
Digg-14	94.9	89.6	93.9	88.8	89.2
Digg-15	88.0	87.8	86.9	80.9	78.9
Digg-16	66.1	88.4	82.5	73.7	61.2
Digg-17	73.4	86.9	87.3	74.0	79.0
Digg-18	74.0	96.9	88.5	77.8	71.3
Digg-19	66.3	84.7	85.3	64.4	55.8
Digg-20	68.4	82.1	74.9	55.3	60.1
Reddit-1	73.2	85.7	89.5	75.4	67.4
Reddit-2	89.9	89.9	93.1	74.1	74.1
Reddit-3	85.2	85.2	90.6	78.0	66.7
Reddit-4	64.6	88.1	88.1	65.2	62.0
Reddit-5	74.2	73.2	87.7	74.2	75.9
Reddit-6	71.4	83.6	82.2	82.2	80.7
Reddit-7	70.0	83.7	88.5	67.6	61.4
Reddit-8	79.4	91.8	79.4	79.4	77.2
Reddit-9	83.2	88.5	91.8	75.6	50.5
Reddit-10	65.0	79.8	76.6	79.8	73.0
Reddit-11	63.4	75.5	67.7	67.7	73.6
Reddit-12	67.6	85.8	91.5	70.2	79.3
Reddit-13	65.4	65.4	87.4	66.4	53.6
Reddit-14	87.4	87.4	87.6	76.9	74.3
Reddit-15	74.9	91.5	89.2	89.2	83.5
Reddit-16	67.6	90.1	90.9	50.5	50.0
Reddit-17	80.9	80.9	81.9	81.9	66.8
Reddit-18	66.2	90.7	82.4	60.9	54.3
Reddit-19	82.4	90.0	79.3	79.3	77.0
Reddit-20	84.6	91.1	84.6	70.7	62.4
MAP	**73.6**	**86.0**	**86.5**	**73.5**	**69.5**

6.4. Case Study and Error Analysis

In this section, we first report a case study by ranking comments for a particular post and see how the rankings change when different techniques for computing similarities are applied. Here, we still set the number of topics to 10, t_1 to 10%, t_2 to 20%, t_3 to 50%, and t_4 to 90% when LDA is applied (similar results are obtained by using HDP). We first look at the following comment:

Table XIV. MAP with Different Threshold Values

$t1$,$t2$	$t3$	$t4$	MAP of Cosine (%)
0.10	0.20	0.30	87.9
0.10	0.30	0.40	88.7
0.10	0.40	0.50	88.7
0.10	0.50	0.60	88.5
0.15	0.25	0.35	88.4
0.15	0.35	0.45	88.9
0.15	0.45	0.55	88.9
0.15	0.55	0.65	88.0
0.20	0.30	0.40	88.9
0.20	0.40	0.50	89.1
0.20	0.50	0.60	89.0
0.25	0.35	0.45	88.9
0.25	0.45	0.55	89.1
0.25	0.55	0.65	88.5
0.30	0.40	0.50	89.1
0.30	0.50	0.60	89.2
0.35	0.45	0.55	89.2
0.35	0.55	0.65	88.7
0.40	0.50	0.60	88.9
0.45	0.55	0.65	88.0
Average MAP			**88.7**

(1) "Short and to the point. Couldn't agree more. I hope that poor woman pulls through."

It is written under a post with the title "The President's statements on the attack in Arizona," which is about President Obama's statement on the attack on congresswoman Gabrielle Giffords in Arizona. The comment is posted to reply to the post directly and is considered as a nondiversionary comment, as its topic is around the congresswoman. There are 100 comments under this post, and 39 of them are considered as diversions. We rank comments in descending order of being diversionary.

When we use term frequency to build document vectors, compute similarities, and then rank comments based on those computations, the comment ranks at the 1st position, as it does not share any common words with the post, which makes its similarity with the post equal to 0. When coreference resolution is applied, the comment's ranking does not change; when extraction from Wikipedia is applied, although more words are added into the post content, the comment is still ranked at the 1st position, as it does not share any words with the post content. When both coreference resolution and extraction from Wikipedia are applied, the comment's rank position stay the same. So far, the comment is still found to be diversionary.

When we build an LDA model on the training data and then infer the document's topic distribution for each comment and the post, the comment is now ranked at the 65th position, as the words "hope" and "woman" in the comment have high probabilities to share the same topic with words such as "Arizona," "tragedy," and "congress," which are all about the post content. When coreference resolution and extraction from Wikipedia are applied to combine with the LDA inference on the test data, the comment is ranked at the 79th position. Now the comment is found to be nondiversionary.

A second comment example is from the post "The risky rush to cut defense spending," which mainly talks about issues on cutting defense spending. There are 102 comments under the post, and 41 of them are labeled as diversions.

(2) "SS brings in more than it pays out. I assume if you cut SS benefits you'll also cut SS tax? Which presents another problem the people who are drawing today paid in years ago, before the tax cut. You really can't cut SS benefits for exactly this reason people paid into the system on the promise that they could draw out when they retire. If we now say they can't draw from it (or can't draw as much from it as they thought) that would be tantamount to default."

And its reply-to comment is this: "Defense spending and Social Security both need reduced. Old people like to feel safe and they like their free money. Therefor Defense spending nor Social Security will never be reduced."

The comment (2) is about social security, which clearly is a diversion from the post content. In addition, its reply-to comment mentions both "defense spending" and "social security," whereas comment (2) chooses to divert the topic to "social security" only, which is not the topic discussed in the post.

When we use term frequency to build document vectors, compute similarities between comments and the post, and between comments and their reply-to comments, then rank comments based on Algorithm 1 (described in Section 5.1.6), comment (2) ranks at the 87th position. It shares a significant number of words with the post content, such as "bring," "pay," and "cut," as a consequence, its similarity with the post is bigger than the threshold t_3 and is put into the PNDL (described in Algorithm 1), although its similarity to the reply-to comment is not high, as they only share one word, "people." When coreference resolution is applied, the comment's ranking does not change; when extraction from Wikipedia is applied and related terms for the proper nouns in the post and comments are added into the post content and some comments, this comment's position moves to the 86th position. When both coreference resolution and extraction from Wikipedia are applied, the comment's rank position stays the same as the ranking position of applying only extraction from Wikipedia. At this moment, the comment is found to be nondiversionary.

When the LDA inference is applied, the post and all comments are represented by the document's topic distributions, and this comment is now ranked at the 39th position. The comment is highly related to the topic with such top terms as "ss," "government," "pay," and "retire," whereas the probability of assigning the post into this topic is very low. Therefore, the similarity between the comment and the post is low. In addition, its similarity to the reply-to comment is not high either, because the reply-to comment is also equally related to other topics, such as the topic with top terms "defense," "cut," and "spend." When coreference resolution and extraction from Wikipedia are applied to combine with the LDA inference, the information in this comment is not expanded by these techniques, but similarities between the post content and some other nondiversionary comments become bigger, and therefore this comment's rank moves up to the 12th position. Now the comment is found to be diversionary.

However, our proposed system is not perfect. A typical nondiversionary comment that is identified as a diversion by our method is a short comment without any pronouns, proper nouns, and topic tokens, as illustrated by the following example. Consider the post discussing Facebook popularity decline among teenagers, in which a reader wrote the following comment: "What's taking over?" It was written to reply to a previous comment: "It's rapidly declining in popularity with my age group also (30ish)." By reading this comment and connecting it to the post, we realize that it is related and nondiversionary; however, the technique that we proposed would treat it as a diversionary comment, as its similarity to the post and to its reply-to comment are both low. In our proposed system, the coreference resolution did not help, as there are no pronouns inside this comment; the extraction from Wikipedia did not help to enrich the context either, as there are no proper nouns in this comment. In addition, the word tokens

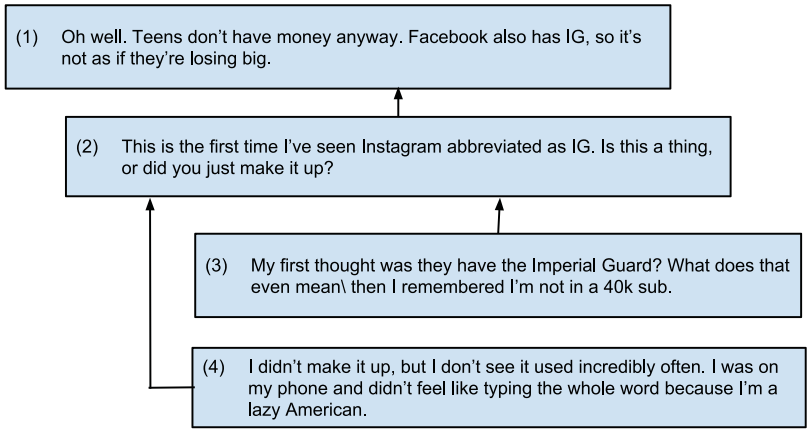

Fig. 6. Comments example under the post about Facebook popularity decline among teens.

of this comment (e.g., take) are not topic indicative, and therefore it is hard to get a reasonable topic distribution by using topic model methods.

There are also diversionary comments that are wrongly identified as nondiversions by our method. Comment (2) in Figure 6 is such a case. In this figure, we show a few comments that are under the post about Facebook popularity decline among teenagers. Comment (1) is related to the post, as it is arguing that there is no need to worry about the popularity decline, as Facebook owns IG (Instagram), which is popular among teens. Comment (2) replies to comment (1); however, it diverts the topic to discuss whether IG is an abbreviation of Instagram. And the following comments (comments (3) and (4)) all follow this diversionary topic. Unfortunately, our system treats comment (2) as a nondiversionary comment. In our approach, by extracting information for "Instagram" from Wikipedia, the content of comment (2) is enriched by keywords such as "social network" and "Facebook," which results in more common words between comment (2), comment (1), and the post content. When applying topic model methods, their similarities are further increased, as the keywords "Instagram," "social network," and "Facebook" allow comment (1), comment (2), and the post to be assigned into the same topic with very high probability.

7. CONCLUSIONS

We presented a study on identifying diversionary comments under blog posts that are prevalent based on our evaluation. In our evaluation dataset, 30.7% of comments were annotated as diversions. Considering that it is difficult to predict whether a reader wants to read an off-topic comment, we suggest that diversionary comments are flagged so that it is up to the reader to decide whether it is worth reading the comments. To the best of our knowledge, this problem has not yet been researched in the literature. We first identified five types of diversionary comments and then introduced rules to determine what a comment replies to under a hierarchy of the post and its associated comments. It then proposed a method to compute the relatedness between a comment and the post content, and the relatedness between a comment and its reply-to comment, which involves coreference resolution, extraction from Wikipedia, and topic modeling (LDA or HDP). Finally, it classified the comments into the categories of diversion or nondiversion, or ranked comments in descending order of being diversionary. The proposed method was evaluated on 4,179 comments from Digg and Reddit. The annotations were done by different annotators, and the agreement of the annotation results

was reported based on Cohen's κ agreement scores. We demonstrated the effectiveness of the proposed method using the MAP measure and the F-measure. Comparisons to baseline methods showed that the proposed method outperformed them considerably. A sensitivity study of different parameter settings was also conducted. The results showed that the parameters performed very well under a large range of values. A future research problem is to identify the different subtypes of diversionary comments.

ACKNOWLEDGMENT

We thank Dr. Tom Moher of the University of Illinois at Chicago for providing advice for conducting the user study.

REFERENCES

Erik Aumayr, Jeffrey Chan, and Conor Hayes. 2011. Reconstruction of threaded conversations in online discussion forums. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 2008. *Modern Information Retrieval* (2nd ed.). Addison-Wesley.

Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. 2007. Clustering short texts using Wikipedia. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*. ACM, New York, NY, 787–788.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*. 294–303. http://dl.acm.org/citation.cfm?id=1613715.1613756

Archana Bhattarai, Vasile Rus, and Dipankar Dasgupta. 2009. Characterizing comment spam in the blogosphere through content analysis. In *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security (CICS'09)*.

Christopher M. Bishop. 2007. Pattern Recognition and Machine Learning. Information Science and Statistics Series. Springer.

Enrico Blanzieri and Anton Bryl. 2008. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review* 29, 1, 63–92. DOI:http://dx.doi.org/10.1007/s10462-009-9109-6

David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022.

Carlos Castillo and Brian D. Davison. 2010. Adversarial Web search. *Foundations and Trends in Information Retrieval* 4, 5, 377–486.

Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. 2006. A reference collection for Web spam. *SIGIR Forum* 40, 11–24.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46.

Gordon V. Cormack. 2008. Email spam filtering: A systematic review. *Foundation and Trends in Information Retrieval* 1, 4, 335–455. DOI:http://dx.doi.org/10.1561/1500000006

Gordon V. Cormack, José María Gómez Hidalgo, and Enrique Puertas Sánz. 2007. Spam filtering for short messages. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. ACM, New York, NY, 313–320. DOI:http://dx.doi.org/10.1145/1321440.1321486

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning* 37, 3, 277–296. DOI:http://dx.doi.org/10.1023/A:1007662407062

Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *Proceedings of the International Symposium on Information Theory (ISIT'04)*.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI'07)*. 1606–1611. http://dl.acm.org/citation.cfm?id=1625275.1625535

Tom Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101, 1, 5228–5235.

Gregor Heinrich. 2004. *Parameter Estimation for Text Analysis*. Technical Report. University of Lepzig, Germany.

Liangjie Hong and Brian D. Davison. 2010. Empirical study of topic modeling in Twitter. In *Proceedings of the 1st Workshop on Social Media Analytics (SOMA'10)*. ACM, New York, NY, 80–88.

Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. 2009. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, NY, 919–928.

Molly Ireland, Amy Gonzales, James W. Pennebaker, Cindy K. Chung, and Roger J. Booth. 2007. *The Development and Psychometric Properties of LIWC2007*. LIWC.net, Austin, TX.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM'08)*. ACM, New York, NY, 219–230. DOI: http://dx.doi.org/10.1145/1341531.1341560

Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, New York, NY, 1549–1552. DOI: http://dx.doi.org/10.1145/1871437.1871669

Solomon Kullback. 2008. *Information Theory and Statistics*. Wiley.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*. 33, 1, 159–174.

François Mairesse, Marilyn A. Walker, Matthias R. Mehl, and Roger K. Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research* 30, 1, 457–500. http://dl.acm.org/citation.cfm?id=1622637.1622649

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY.

Juan Martinez-Romo and Lourdes Araujo. 2009. Web spam identification through language model analysis. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'09)*. ACM, New York, NY, 21–28. DOI: http://dx.doi.org/10.1145/1531914.1531920

Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. 2012. Adding semantics to microblog posts. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12)*. ACM, New York, NY, 563–572.

Gilad Mishne. 2005. Blocking blog spam with language model disagreement. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05)*.

Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam Web pages through content analysis. In *Proceedings of the 15th International Conference on World Wide Web*.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (HLT'11)*. 309–319. http://dl.acm.org/citation.cfm?id=2002472.2002512

Gerard Salton, Andrew Wong, and Chungshu S. Yang. 1974. *A Vector Space Model for Automatic Indexing*. Technical Report. Ithaca, NY.

David Sculley and Gabriel M. Wachman. 2007. Relaxed online SVMs for spam filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*. ACM, New York, NY, 415–422. DOI: http://dx.doi.org/10.1145/1277741.1277813

Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. In *Handbook of Latent Semantic Analysis*, T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch (Eds.). Lawrence Erlbaum Associates, 427–448.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2004. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101, 476, 1566–1581.

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 the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. 2011. Linking online news and social media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM'11)*. ACM, New York, NY, 565–574.

Dan Twining, Matthew M. Williamson, Miranda J. F. Mowbray, and Maher Rahmouni. 2004. Email prioritization: Reducing delays on legitimate mail caused by junk mail. In *Proceedings of the USENIX Annual Technical Conference (ATEC'04)*. 4. http://dl.acm.org/citation.cfm?id=1247415.1247419

Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han. 2011a. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 435–444.

Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. 2011b. Predicting thread discourse structure over technical Web forums. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 13–25.

Yi-Min Wang, Ming Ma, Yuan Niu, and Hao Chen. 2007. Spam double-funnel: Connecting Web spammers with advertisers. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM, New York, NY, 291–300. DOI: http://dx.doi.org/10.1145/1242572.1242612

Tae Yano, William W. Cohen, and Noah A. Smith. 2009. Predicting response to political blog posts with topic models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'09)*. 477–485.

Mingliang Zhu, Weiming Hu, and Ou Wu. 2008. Topic detection and tracking for threaded discussion communities. In *Proceedings of the 2008 IEEE / WIC / ACM International Conference on Web Intelligence and Intelligent Agent Technology, Volume 01*. IEEE, Los Alamitos, CA, 77–83.

Li Zhuang, John Dunagan, Daniel R. Simon, Helen J. Wang, and J. Doug Tygar. 2008. Characterizing botnets from email spam records. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*. Article No. 2. http://dl.acm.org/citation.cfm?id=1387709.1387711

Received July 2013; revised March 2015; accepted June 2015