Content-Aware Trust Propagation Toward Online Review Spam Detection

HAO XUE, QIAOZHI WANG, BO LUO, HYUNJIN SEO, and FENGJUN LI,
The University of Kansas, USA

With the increasing popularity of online review systems, a large volume of user-generated content becomes available to help people make reasonable judgments about the quality of services and products from unknown providers. However, these platforms are frequently abused since fraudulent information can be freely inserted by potentially malicious users without validation. Consequently, online review systems become targets of individual and professional spammers, who insert deceptive reviews by manipulating the rating and/or the content of the reviews.

In this work, we propose a review spamming detection scheme based on the deviation between the aspect-specific opinions extracted from individual reviews and the aggregated opinions on the corresponding aspects. In particular, we model the influence on the trustworthiness of the user due to his opinion deviations from the majority in the form of a deviation-based penalty, and integrate this penalty into a three-layer trust propagation framework to iteratively compute the trust scores for users, reviews, and review targets, respectively. The trust scores are effective indicators of spammers, since they reflect the overall deviation of a user from the aggregated aspect-specific opinions across all targets and all aspects. Experiments on the dataset collected from Yelp.com show that the proposed detection scheme based on aspect-specific content-aware trust propagation is able to measure users' trustworthiness based on opinions expressed in reviews.

CCS Concepts: • **Security and privacy** → *Social aspects of security and privacy*;

Additional Key Words and Phrases: Spam detection, social networks, trust propagation, opinion mining

ACM Reference format:
Hao Xue, Qiaozhi Wang, Bo Luo, Hyunjin Seo, and Fengjun Li. 2019. Content-Aware Trust Propagation Toward Online Review Spam Detection. *J. Data and Information Quality* 11, 3, Article 11 (June 2019), 31 pages.
https://doi.org/10.1145/3305258

1 INTRODUCTION

The increasing popularity of online collaborative platforms has tremendously facilitated online interactions and information sharing. As a result, a vast amount of user-generated content (UGC) has been made available online. For example, *TripAdvisor.com*, which specializes in travel-related services, has reached 315 million unique monthly visitors and over 200 million reviews. *Yelp.com*, which is known for restaurant reviews, has a total of 71 million reviews of businesses and a

This work is sponsored in part by the University of Kansas under INS-0073037, the National Security Agency (NSA) SoS Initiative, and the National Science Foundation under NSF CNS-1422206 and DGE-1565570.

Authors' addresses: H. Xue, Q. Wang, B. Luo, and F. Li (corresponding author), 2001 Eaton Hall, University of Kansas, 1520 West 15th Street, Lawrence, KS 66045; emails: {haoxue, qzwang, bluo, fli}@ku.edu; H. Seo, 207A Stauffer-Flint Hall, University of Kansas, 1435 Jayhawk Blvd., Lawrence, KS 66045; email: hseo@ku.edu.

1936-1955/2019/06-ART11 $15.00
https://doi.org/10.1145/3305258

monthly average of 135 million unique visitors to the site. This plethora of data provides a unique opportunity to the formation of “aggregated opinions,” which help people to make reasonable judgments about the quality of service/product of an unknown provider. However, since such platforms can be easily abused, the quality of UGC is often problematic. A non-negligible portion of online reviews is unfairly biased or misleading. To make things worse, deceptive reviews have been purposely inserted into the online review systems by individual or professional spammers [9, 22, 34, 72]. A recent study on Yelp [38] showed that about 16% of restaurant reviews are flagged by Yelp’s review filter as “not recommended.”

With a goal to mislead people and manipulate their decisions, opinions expressed in such reviews often deviate significantly from the fact. This is known as online *review spamming*. Jindal et al. first identified this problem in [22]. Compared with other forms of spamming, such as web spams and email spams, the review spamming is much more difficult to detect. The content of spam reviews does not contain clear clues about spamming, such as a malicious URL or a malicious attachment. Therefore, it is difficult to establish the “ground truth” that distinguishes spam reviews from normal reviews [45, 52, 71, 72]. Some studies recruited people from Amazon Mechanical Turk to create synthetic spam reviews. However, these approaches were criticized because the quality of the tasks completed on crowd-sourcing platforms is often lower than expected and the selected Turkers’ behavior may not resemble spammers’.

Review spamming detection is a challenging problem. To address this problem, many online review systems adopt deterrence-based or reputation-based detection approaches. For example, the “Verified Purchase” mechanism of *Amazon.com* allows only customers who actually made the purchase to post reviews. A more generally adopted approach is the “review of the review,” which allows users to rate a review or vote for its “helpfulness,” and uses the ratings or vote counts as a measure to assess review quality. While these mechanisms provide additional information about how helpful or trustworthy a review is, their limitations are also obvious. Similar to reviews, the “review of review” is a subjective judgment and itself can also be faked. Moreover, it often suffers from inadequate user participation. Surveys show that only a small portion of users provides reviews online. Furthermore, among thousands of views of a review, only a few readers provide feedbacks. As a result, a large amount of reviews has no helpfulness or usefulness rating at all.

On the other hand, detection-based mechanisms are considered more suitable since spamming activities often demonstrate certain patterns. Many learning-based schemes have been proposed to identify deceptive reviews and reviewers from textual features [22, 47, 52], temporal features [9, 77], spammers’ individual or group behavior patterns [32, 47, 72], and sentiment inconsistency [49]. The rationale behind these approaches is twofold. Along the first direction, detection models rely on the deviation in rating behaviors. Since the objective of opinion spammers is to alter users’ perception of the quality of the target, they often generate a large amount of reviews with extreme ratings using different accounts. In this way, spammers can significantly distort the mean rating. So, the detection focuses on rating-based features that reflect the deviation from the aggregated rating (or the majority view) [1, 34] and other rating behaviors (e.g., change of average rating over time, change of average ratings across groups of users). While these approaches have been used with success, they can be easily gamed by spammers who evolve to avoid extreme rating behavior. Along the second direction, detection schemes rely on text features, such as duplicated texts [22, 47], psycholinguistic deceptive characteristics [52], and so forth. These approaches use spam reviews that are manually identified or created to train the classifiers, which introduce expensive labeling costs. To make things worse, automated spam review generators based on deep learning have been proposed recently [80], which leverage text features used in spam detection for spam generation.

In this work, we propose to use the inconsistency in opinions expressed in one's review and the consensus opinions shared by a group of anonymous users to determine the trustworthiness of a review and its reviewer. This is inspired by the approaches that identify opinion spams whose ratings are inconsistent with the opinions expressed in the review text [34, 49]. Moreover, we target spams that are carefully crafted to evade existing detection mechanisms, rather than the ones with simple duplication or obvious deceptive cues. An important assumption in our approach is that *in a healthy review system, benign users should always outnumber spammers*, which is also the assumption of any reputation-based systems. Therefore, the aggregated opinion from a majority vote should reflect the actual quality of service/product on every aspect. Although the majority views have some limitations in extreme cases, such as inertia against sudden change of quality, we assume that in most cases the majority view reflect the facts effectively.

Our scheme takes multiple deviation indicators into consideration and integrates the deviations across all aspects to obtain an overall deviation. In particular, we present two approaches to extract aspect-specific opinions. In the first approach, we trained a classifier with a small labeled dataset using supervised learning [78]. While achieving a high accuracy, it is costly due to data labeling and rigid as the aspect categories need to be predetermined. To overcome the limitations, we developed an unsupervised approach to extract aspects from review content. To effectively integrate the aspect-specific indicators, we adopt a three-layer trust propagation framework, which was first proposed in [70], to calculate trust scores for users, reviews, and the aspect-specific opinions of the target (called "statement"). Using the extracted opinions as input, the three-layer trust propagation framework iteratively computes the trust scores and propagates them among users, reviews, and statements. The converged trust score reflects the overall deviation of a user from the aggregated opinion across all the aspects and all the reviewed targets, which we believe is a strong indicator of trust to distinguish regular users and spammers.

The contributions of this article are threefold: (I) We propose a novel aspect category-specific opinion indicator as a content-based measure to quantify the quality and trustworthiness of review content. We focus on detecting carefully composed opinion-wise deviated spam reviews, rather than typical spam reviews that can be easily detected using existing behavior-based or rating-based methods. (II) We propose a three-layer trust propagation model based on the intertwined relationships between three types of nodes: users, reviews, and statements. Compared to other link-based approaches [72] that utilized behavioral features such as rating deviation, our system takes opinions of multiple aspect categories extracted from review content into account, which captures the opinions expressed by users directly. And (III) we develop an effective iterative computational framework for three concepts that model the level of trustworthiness of the three types of nodes, namely, honesty of users, faithfulness of reviews, and truthfulness of statements. Compared to iterative frameworks like [70, 72], our framework is not only based on the reinforcements between the three concepts, but also on the level of consensus between individual opinion and aggregated opinion.

The rest of the article is organized as follows. Section 2 introduces the problem and provides an overview of our solution. We define aspect-specific opinion indicators in Section 3 and present the three-layer trust propagation framework in Section 4, followed by experiments and evaluations in Section 5. Finally, we review the related works in Section 6 and conclude in Section 7.

2 THE PROBLEM AND THE OVERVIEW OF OUR SOLUTION

Rating deviation is widely used in previous review spamming detection approaches. It is a strong indicator of spamming, since the primary goal of the spammers is to promote or demote the target entity by increasing or decreasing its average rating dramatically. However, rating deviation-based detection scheme can be evaded. For example, the spammers can regulate their behavior by avoiding inserting extreme ratings within a short period to change their temporal rating patterns. To tackle this problem, we propose an opinion deviation-based trust indicator as a new detection feature, which is robust to detection evasion.

Obviously, in an online review, the opinion of users is expressed not only in the rating of the review, but also in the review content. Some previous approaches compare the sentiment extracted from the content of a review with its rating to look for inconsistent opinions, and use it to detect spam reviews. However, since conflicting opinions may be expressed in the reviews, these approaches are limited due to the poor performance of sentiment analysis especially when conflicts exist. Therefore, in this work, we propose to treat a review as a set of ratable aspects with corresponding sentiments. Since conflicting opinions in a review often regard different aspects of the target entity, by dividing the review into sets of words regarding different aspects, our approach can effectively address this problem. To do so, we first conduct opinion mining on the content of each review to extract the opinions on a set of aspects. In particular, we propose a supervised-learning-based approach and an unsupervised-learning-based approach to extract opinions across a set of aspects from a dataset of reviews collected from Yelp.com. Since the extracted opinions are on multiple aspects, we can group opinions from different users on the same aspect of the target and compute the aggregated opinion, which highly likely reflects the "fact" (e.g., the actual quality of the reviewed entity on this aspect). Therefore, the deviation of a user's opinion from the aggregated opinion may reflect her trustworthiness.

To measure the influences on users' trustworthiness due to opinion deviation, we propose a three-layer trust propagation framework to compute the overall influences across multiple entities and aspects. The higher the score is in the final output, the more trustworthy an entity is (user, review, or statement); the details are discussed in Section 4. As shown in Figure 1, u_i represents a user, r_i represents a review, and e_i represents an entity (e.g., a restaurant). Given an application domain, users may be interested in only a few abstract *aspects categories* about the targets. For example, in restaurant reviews, users are more interested in the food flavor and quality, price, service, atmosphere, and so forth. In product reviews, users are more concerned about quality, lifetime, price, and so forth. Each aspect category may consist of more specific aspects, which can be directly extracted by our opinion mining algorithm. We use $ac_{i,j}$ to represent the j-th aspect category of entity e_i. Then, we construct *statements* for each entity. Each statement is an opinion on a particular aspect category of the entity, for example, "restaurant1-food-positive" is a statement that expresses the "positive" opinion toward the aspect category "food" of entity "restaurant1."

For ease of presentation, we list the notations of terms and their meanings in Table 1. Some of the terms will be defined in later sections.

3 ASPECT CATEGORY SPECIFIC OPINION INDICATOR

Existing content-based detection approaches take textual content of a review as input, which often use word-level features (e.g., n-grams) and known lexicons (e.g., WordNet [43] or psycholinguistic lexicon [34]) to learn classifiers that identify a review as spam or non-spam. To train the classifier, costly and time-consuming manual labeling of reviews is required. Due to the subjectiveness of human judgment and personal preferences, there is no readily available ground truth of opinions. Therefore, a high-quality labeled dataset is difficult to obtain. Some existing works adopt crowdsourcing platforms such as Amazon Mechanical Turk to recruit a human labeler; however, it is pointed out that the quality of the labeled data is very poor. Different from these approaches, our opinion spam detection scheme utilizes the deviation of the majority opinion. Although biased opinions always exist in UGC, we argue that a majority of users may be *biased but honest*, instead of maliciously deceptive. This is based on an overarching assumption regarding reviewer behaviors—that is, the majority of reviews are posted by honest reviewers, as recognized by many existing

Table 1. Notations of Terms and Concepts

Notation	Definition
u	a user
r	a review
s	a statement
e	an entity
a_i	the i-th aspect word
ac_i	the i-th aspect category
l_i	the i-th sentiment label
$ao_{u,e}$	the aggregated opinion of user u to entity e
o	an opinion vector
os	an opinion status vector
$h^{(n)}(u_i)$	honesty score of user u_i in round n
$f^{(n)}(r_i)$	faithfulness score of review r_i in round n
$t^{(n)}(s_i)$	truthfulness score of statement s_i in round n
$\delta^{(n)}(u_i)$	average deviation of user u_i in round n
$\xi(u_i, r_j)$	the confidence of u_i on r_i about the target entity
$\rho(r_i, s_i)$	the relevance of r_i to s_i

works on opinion spam detection [1, 34, 49]. A news report [50] states that it is estimated that among online hotel reviews, between 1% and 6% of positive reviews are fake. A recent study shows that about 16% of the restaurant reviews on Yelp are filtered by Yelp's filter [38]. The fact that the functionality of the online review systems depend on the well-being of systems themselves make it barely possible for spammers to dominate the review systems. We argue that if this assumption does not hold, online peer review systems will be completely broken and useless. Furthermore, hiring a huge number of spammers to dominate the opinions of the review systems is very costly and infeasible. As a result, we propose to use the majority opinions as the "ground truth."

3.1 Aspect-Based Opinion Extraction

Existing work [13, 18, 40, 46, 55] on opinion mining studies opinions and sentiments expressed in review text at document, sentence, or word/phrase levels. Typically, the overall sentiment or subjectiveness of a review (document-level) or a sentence of a review is classified and used as a text-based feature in spam detection. However, we consider these opinions are either too coarse or too fine-grained. For example, opposite opinions are commonly expressed in an individual review—it may be positive about one aspect of the target entity but negative about another. This is difficult to capture using the document-level sentiment analysis. Therefore, the derived review-level majority opinion is inaccurate and problematic.

Another direction of approaches proposes to use opinion features that associate opinions expressed in a review with specific aspects of the target entity [16, 49]. Intuitively, these opinion features are nouns or noun phrases that typically are the subjects or objects of a review sentence. For example, in the below review, the underlined words/phrases can be extracted as opinion features.

"*This place is the bomb for milkshakes, ice cream sundaes, etc. Onion rings, fries, and all other "basics" are also fantastic. Tuna melt is great, so are the burgers. Classic old school diner ambiance. Service is friendly and fast. Definitely come here if you are in the area ...*"

In some cases, users may comment on a large number of specific aspects about the target entity. The derived opinion features are too specific and too fine-grained to form a majority opinion on each feature, since other reviews about the same target may not comment on these specific features. However, from the above example, we can see that opinion features such as "milkshakes," "fries," and "burgers" are all related to an aspect category "food." If we define a set of aspect categories, opinion features about a same or a similar high-level concept can be grouped together.

Consider a set of reviews (**R**), which are written by a group of users (**U**) about a set of entities (**E**). Each review $r \in \mathbf{R}$ consists of a sequence of words $\{w_1, w_2, \ldots, w_{n_r}\}$. Then, we can define a set of m aspect categories $ac = \{ac_1, ac_2, \ldots ac_m\}$, where each aspect category ac_i covers a set of aspects a_i that represent the same concept. For each aspect, correspondingly there is a sentiment polarity label $l = \{l_1, l_2, \ldots l_k\}$. As a result, for each review r, we can extract a set of aspect-sentiment tuples $<a_i, l_i>$. By aggregating the tuples that belong to the same aspect category together, we get the the aggregated opinion for each aspect category $<ac_i, l_i>$. Combining aggregated opinions for all commented aspect categories, a final aggregated opinion $ao_{u,e} = \{<ac_i, l_i>\}$ is used to represent the aspect category-specific opinions of a user u toward a target entity e.

Typical sentiment polarity labels include "positive," "negative," "neutral," and "conflict" [13, 58]. The "conflict" label captures inconsistencies within a review but does not provide any information about the inter-review consistency, therefore we do not consider this label in our model.

In this work, we use Yelp reviews as our dataset to study the credibility of users and their reviews. We present two approaches for opinion extraction: supervised learning based and unsupervised learning based. We first started with supervised learning methods as it is fast and simple. With data labeled with aspect categories and corresponding sentiment, identifying the aspect-sentiment tuples is quite straightforward. However, its limitations are also obvious. First, getting data labeled requires human labor and is time-consuming. Second, the number and type of categories are predefined due to the labeled data and the type of data can be applied on is also limited. The effort of adding a new category is often costly as it usually requires re-labeling the data. On the other hand, with unsupervised learning, we gain the flexibility of identifying opinions expressed in more aspect categories and the entire opinion mining process can be automated with minimal human intervention. Although the unsupervised method may cause some loss of accuracy in terms of aspect category compared to the supervised method, we can tolerate some inaccurate classifications as our proposed framework does not rely on output from a single category. Note that the output of these two types of methods is different, thus the results of the two approaches cannot be combined.

3.2 Supervised Opinion Extraction

The study in [13] identifies six categories for restaurant reviews, *food, price, service, ambience, anecdotes*, and *miscellaneous* for review classification. We followed up the idea and define a small set of aspect categories including four meaningful aspects *food, price, service*, and *ambience* for restaurant reviews. We treat the opinion extraction as a classification problem and adopt the Support Vector Machine (SVM) supervised learning model for opinion extraction. We use the SemEval dataset [58], which is a decent-sized set of labeled data for restaurant reviews, to train our classifier (see more details in Section 5). Our goal is to identify an adequate number of aspects that are commonly expressed in reviews so that we can construct a credibility indicator from the aggregated opinions. In fact, too many over-specific aspects complicate the credibility computing model instead of improving it. For example, consider the review below:

"*This place is the bomb for milkshakes, ice cream sundaes, etc. Onion rings, fries, and all other "basics" are also fantastic. Tuna melt is great, so are the burgers. Classic old school diner ambiance. Service is friendly and fast. Definitely come here if you are in the area ...*"

The words with an underline are aspects that belong to the category "food." If we consider every single ratable aspect word as an aspect category, too many categories will exist. It is helpful to construct the aspect-specific indicators if we consider aspects that represent the same concept separately. Therefore, we mainly focus on the four high-level aspect categories rather than more specific aspect categories.

Besides the four major categories, we combine all other aspect category labels as a fifth category, *miscellaneous*. Also, using *miscellaneous* also helps improve the quality of classifications of the first four categories (i.e., *food, price, service*, and *ambience*) as the classifier will not misclassify some irrelevant aspects into those four categories. We first applied the trained classifier on reviews with sentence-level to identify the aspect categories each sentence is about. Note it is possible that a sentence is about multiple aspect categories. For example, the sentence *The burger here is pretty good but the price is a bit expensive* talks about two aspect categories, *food* and *price*.

Next, we conduct the aspect-specific sentiment classification upon the classified aspect-specific sentences to obtain aspect-specific sentiment polarities. For each category, the classification is conducted independently. For example, to determine the sentiment polarities of the "food" category, we conduct a sentiment classification upon all sentences that have been classified into the category "food," and determine the aspect-sentiment tuples: "food-positive," "food-negative," and "food-neutral."

3.3 Unsupervised Opinion Extraction

While the supervised approach can directly identify the aspect categories from a sentence or a review, the approach using unsupervised learning consists of several steps.

Opinion Extraction with Dependency Relations. Rather than directly assigning classification labels to a review, the aspect-specific opinions need to be explicitly identified. As mentioned earlier, the objects (i.e., the aspect a_i in the tuple) of an opinion feature are usually expressed as nouns or noun phrases. On the other hand, the modifiers (i.e., the sentiment label l_i in the tuple) are usually expressed as adjectives, adverbs, or verbs with sentiment orientations. Thus, the aspect-sentiment tuple can be identified to search for certain patterns of POS tags. However, the object and corresponding modifier may not necessarily occur close to each other in a sentence. Manually defining POS tag patterns can be costly and inaccurate. Thus, in this work, we used dependency relations [41] to parse and extract the qualified expressions.

A dependency relation is an asymmetric binary relationship between a term called head or governor and another term called modifier or dependent [41]. As suggested in [76, 79], we decided to use three types of dependencies, including "nsubj," "amod," and "dobj" in the opinion extraction process, denoting subject-predicate relations, adjectival modifying relations, and verb-object relations, respectively.

Aspect Categorization. Unlike supervised method, in which labels of aspect categories are directly assigned by the classifier, unsupervised method requires to group the extracted aspects that are semantically similar into aspect categories. There are many ontology-based lexical tools like WordNet [43] that provide the synonyms of words. However, the limitation of these tools is that the concept-based synonyms provided are predefined and fixed. Also, a word usually has more than one "sense" (treated as polysemy), calculating semantic similarity needs to find the correct sense manually, which cannot be processed in an automatic manner. Another difficulty is that the semantic similarities sometimes only exist in a certain context, for example, "steak" and "egg" are semantically similar only in the context of restaurant reviews. Without the context, "steak" is more related with meat while "egg" is more related with bird. In online reviews, this is often the case with ratable aspects. Thus, the semantic similarity cannot be acquired from some predefined tools but needs to be learned directly from the reviews. Topic modeling approaches are able to group words in topics based on textual data, but the topics are usually not coherent enough to be used as aspect categories.

In the field of Natural Language Processing (NLP), the existing work that studies semantic representation of words usually works under the assumption that words occurring within similar contexts are semantically similar [17]. Vector-based models have been successful in representing the semantic relationships among words. Conceptually, mapping words or phrases to vectors are known as word embeddings. Models such as word2vec [42] and GloVe [57] have proved their effectiveness and outperformed simple models in many NLP tasks.

In this work, we used the word2vec model to learn the word embeddings of all words occurred in the reviews. With vector representations of aspects learned from the review, it is much easier to compute the similarities. Intuitively, aspects that occur in similar context will have similar embeddings. To group aspects into categories, we applied K-means, a widely used clustering algorithm that is able to partition the data into k clusters based on feature similarity. We used the learned word embeddings of all aspects as the features for K-means as the word embeddings capture the semantic similarities well enough. The output clusters represent the aspect categories learned based on the semantic similarities captured from the reviews.

Sentiment Orientation of Modifiers. After aspects are categorized, the next task is to assign a sentiment label (i.e., l_i) to the corresponding modifiers extracted from the dependency relations. Although we also obtained the word embeddings of the modifiers using word2vec, we cannot apply a clustering algorithm to group them into different sentiment labels. When word2vec is trained, the sentiment orientation is not embedded in the training process, thus the obtained word vectors do not contain information about the sentiment polarity. Actually, modifiers with opposite sentiment polarities will both co-occur with certain target words frequently. For example, “expensive” and “cheap” are of opposite sentiment polarities in terms of price. However, they may both occur in a similar context. As a result, the similarity between their word vectors is high. If we apply K-means on the word embeddings of the modifiers, “expensive” and “cheap” will be grouped in the same cluster even though they have the opposite sentiment polarity for price.

Instead of utilizing the word embeddings with K-means, we used a widely adopted opinion lexicon [18] to identify positive and negative modifiers. The lexicon contains two lists of words, one with 2,006 positive words and the other with 4,783 negative words. For each extracted modifier, if it is included in one of the lists, we assign the corresponding sentiment label to it. If the modifier is not included in either of the lists, we ignore it. Note, there is no list of words for the sentiment orientation “neutral”; we will only have two types of sentiment labels in this unsupervised-based setting: “positive” and “negative.”

3.4 The Opinion Vector and Quality Vector

To use the extracted opinions for further analysis, we define an opinion vector $\mathbf{o} = [o_1, \ldots, o_5]$ to capture aspect-specific opinions and their sentiment polarities. Each element of the opinion vector corresponds to an aspect of food, price, service, ambience, and miscellaneous, respectively. Sentiment polarities are represented by element values, where a positive sentiment is denoted by “+1,” a negative sentiment is denoted by “−1,” and neutral is denoted by “0” (if provided). Since a statement may not necessarily express an opinion about an aspect, we distinguish no opinion expressed from a neutral opinion by defining a corresponding opinion status vector **os**. For example, if a statement expresses three opinions, positive about food, neutral about price, and negative about service, its opinion vectors are $\mathbf{o} = [1, 0, -1, 0, 0]$ and $\mathbf{os} = [1, 1, 1, 0, 0]$.

With the opinion vectors, we can aggregate the opinions on multiple aspects from all reviewers of an entity to form four aspect-specific aggregated opinions. While aspect-specific opinions are subject judgments and thus can be biased, the aggregated aspect-specific sentiments are highly likely to reflect the true quality of the entity from a specific aspect. This is because individual biases are typically smaller aspect level than at document level, which is more affected by the weights subjectively assigned by individuals to multiple aspects. In this sense, aspect-level bias can be corrected by the majority view if the review amount is adequate. Furthermore, comparing with rating, aspect-specific opinions are more difficult to be tampered by opinion spammers, whose review texts are likely to be pointless, wrongly focused, or brief. Finally, the aggregated sentiments are robust to correct the inaccuracy introduced by opinion mining models. Opinion mining often suffers from precision problems, but our goal is to decide if the overall aspect-specific opinion is positive, neutral, or negative. Although each individual input incurs a small uncertainty, the chance to affect overall value is very small. Based on these considerations, we derive the aggregated aspect category-specific opinion vectors as $\mathbf{o}_{\mathbf{agg}} = [o_{agg_1}, \ldots, o_{agg_5}]$ and $\mathbf{os}_{\mathbf{agg}} = [os_{agg_1}, \ldots, os_{agg_5}]$, where

$$o_{agg_i} = \begin{cases} 1, & avg_{i \in A_i}(o_i) \geq \theta_p \\ -1, & avg_{i \in A_i}(o_i) \leq \theta_n \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In this way, the aggregated sentiment polarity of each aspect is mapped to the positive, neutral (not included with opinions extracted using unsupervised method), and negative labels based on the averages. The aggregated aspect category-specific opinion vector is considered a *quality vector*, which can be used to determine the credibility of a user statement. Intuitively, a statement is more credible and of higher quality, if it expresses a consistent opinion with the aggregated opinion about one aspect of the target entity, and thus the reviewer is considered more honest and trustworthy.

4 CONTENT-BASED TRUST COMPUTATION

We compute the aggregated aspect-specific opinion vector as a quality measure and use the individual aspect-specific opinion vector as a credibility (or trust) measure. To integrate trust measures across multiple users and multiple entities, trust propagation models are commonly used [70, 81]. In this work, we model the relationships and inter-dependencies between user, review, and the entity being reviewed, and adopt a three-layer trust propagation model to compute the trust-related scores for users, reviews, and aspect-specific statements iteratively.

4.1 The Three-Layer Trust Relationships

The three-layer trust propagation model was first introduced in [70] to measure the trustworthiness of online claims and their sources, especially when conflicting information is provided by multiple sources. Traditionally, this problem was modeled as a trust propagation problem using the bipartite graph consisting of *sources* and *evidences* that support a same claim. In particular, the trustworthiness of a source relies on the confidence of all the evidences that it provides to support its claims, and the confidence of a claim depends on the trustworthiness of all sources that provide evidences to it. Different from the bipartite graph-based two-layer models, the three-layer model introduces an additional intermediate layer to represent the influence on one evidence due to another evidence of the same claim. As explained in [70], the inter-evidence interactions can be used to model the similarity between two evidences so that similar evidences receive similar trustworthiness scores.

We adopt the three-layer architecture to model the relationships among three types of nodes (i.e., *users*, *reviews*, and *statements*) in our review system, in which the inter-dependencies between nodes and on the links are completely different from the ones in [70] and thus need to be re-defined. The key idea of adopting the three-layer architecture is to use the intermediate layer to model the

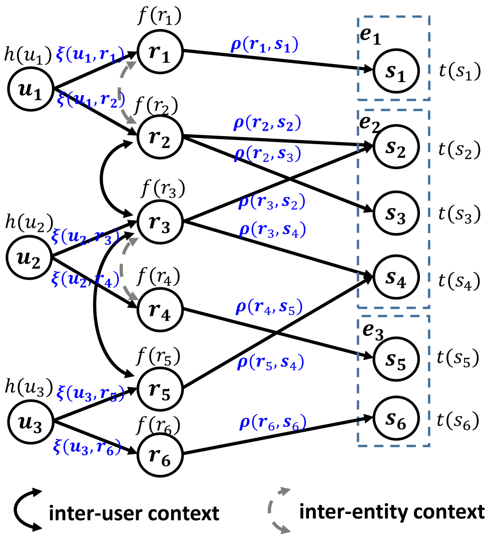


Fig. 1. The three-layer trust propagation model.

inter-review interactions and the influence on one review due to the other reviews about the same entity (or more precisely the same aspect of that entity).

Nodes. As shown in Figure 1, we denote a *user*, a *review* of a user, and a *statement* about a target entity as u_i, r_j, and s_k, respectively. Here, the statement is defined as an opinion expressed on a particular aspect of the target entity. For example, a statement $restaurant_1 - food - positive$ expresses a positive opinion about the food of $restaurant_1$. We then assign trust scores to each node, more specifically, an *honesty* score, denoted as $h(u_i)$, is assigned to each user node, a *faithfulness* score, denoted as $f(r_i)$, is assigned to each review node, and a *truthfulness* score, denoted as $t(s_i)$, is assigned to each statement node. The three types of trust scores take values between 0 and 1.

Edges. The edge from a user node to a review node means the user posts this review, and the edge from a review node to a statement node represents the review supports this statement. As shown in Figure 1, a user can post one review regarding multiple statements of the target entity (e.g., u_2 posts r_3 on s_2 and s_4 of entity e_2) or multiple reviews about multiple entities (e.g., u_1 posts r_1 about e_1 and r_2 about e_2), and multiple users can post reviews on the same aspect of a target entity (e.g., u_1 and u_2 post r_2 and r_3 on s_2 of e_2, respectively). Finally, we use the double-arrow lines to denote the inter-review influences on one review due to other reviews. In particular, the solid double-arrow line depicts the inter-review influence on a review due to reviews from other users on the same aspect of the same target entity, and the dashed double-arrow line represents the inter-review influence on a review caused by reviews from the same user on the same aspect of different entities.

Interactions on Edges. There are four main types of interactions in our three-layer trust propagation model: (1) The user-review edges represent the confidence of the user on the review about the target entity, denoted as $\xi(u_i, r_j)$. Therefore, the faithfulness of a review can be calculated as $f(r_i) = h(u_i)\xi(u_i, r_j)$, where $\xi(u_i, r_j)$ is normalized by the maximal $\xi(u_i, r_j)$ of all reviews posted by u_i. It measures the relative reliability of a particular review among all the reviews posted by a same user who reviews multiple different entities. (2) The review-statement edges represent the relevance of a review to a statement of the target entity, denoted as $\rho(r_i, s_i)$. Opinions expressed in all reviews about a target entity regard multiple aspects of the entity. Given a statement s_i about a particular aspect of the target, $\rho(r_i, s_i)$ measures how relevant the review r_i is about the statement and how strong the sentiment expressed in the review supports the statement. (3) The review-review edges in the inter-user context group reviews from different users on the same aspect of a target entity. This leads to a voting strategy to aggregate specific opinions, in terms of positive, negative, and neutral, on each aspect of the reviewed entity. Thus, the deviation between the individual opinion and the aggregated opinion provides useful information to distinguish normal users and spammers. This is because benign users may express different opinions in some cases due to individual experiences or subjectivity; only spammers continuously express an opposite opinion against the fact. Therefore, in the inter-user context, the influences on a review due to other reviews about the same statement of the target can be measured by its opinion deviation. And (4) the review-review edges in the inter-entity context group reviews from the same user on the same aspect of different targets. It provides a context to measure individual bias toward different aspects. For example, one user may be concerned more about one aspect over the other aspects, so her reviews on this aspect are more significant than her reviews on aspects on which she usually does not comment.

4.2 Trust Propagation and Trust Scores

As described in Section 4.1, three types of trust scores are defined for user, review, and statement nodes. Similar to the approach presented in [70], the three trust scores can be computed iteratively over the trust framework following the interactions between the three types of nodes.

The Basic Trust Propagation Model. In the basic model, we consider only the influences on the trust score of each node due to trust scores of other connected nodes. We start with the *faithfulness* score for the review, which represents the confidence in the trustworthiness of the review. Initially, the *faithfulness* score of review r_i is $f^{(0)}(r_i)$. It can be set to 1, denoting an equal confidence in the trustworthiness of all reviews, or estimated by other spamming detection algorithms based on content, structure, or behavior-based features. We then update the faithfulness scores for all reviews following the below equation:

$$f^{(n+1)}(r_i) = \mu f^{(n)}(r_i) + (1-\mu) h^{(n)}(u(r_i)), \tag{2}$$

where $\mu \in (0,1)$ is an interpolation coefficient that controls the bias of prior knowledge of $f(r)$ on future estimates of the review faithfulness.

Similarly, the *truthfulness* score of a statement can be calculated as

$$t^{(n+1)}(s_i) = \frac{\sum_{r_j \in \mathcal{R}(s_i)} [f^{(n)}(r_j) \times h^{(n)}(u(r_j))]}{|\mathcal{R}(s_i)|}, \tag{3}$$

where $\mathcal{R}(s_i)$ is the collection of all the reviews about statement s_i (on a particular aspect of the target entity). From Equation (3), we see that the truthfulness score of a statement s_i is relevant to the average trustworthiness of all the reviews regarding the aspect expressed in the statement, and the honesty scores of the users whose reviews are relevant to this statement.

Finally, the *honesty* score of a user depends on the average trustworthiness of all the statements that he supports. Therefore, it can be calculated as

$$h^{(n+1)}(u_i) = \frac{\sum_{s_j \in \mathcal{S}(u_i)} t^{(n)}(s_j)}{|\mathcal{S}(u_i)|}, \tag{4}$$

where $\mathcal{S}(u_i)$ is the set of statements about which the user u_i posts reviews.

From the equations above, it is obvious that three types of trust scores are influenced by each other following the edges that connect them and thus can be computed iteratively from an initial setting.

The Enhanced Model with User-Review and Review-Statement Interactions. We improve the basic trust propagation model by integrating the confidence and relevance factors onto the user-review and review-statement edges, respectively. In particular, we update Equations (2) and (3) as below:

$$f^{(n+1)}(r_i) = \mu f^{(n)}(r_i) + (1-\mu)[h^{(n)}(u(r_i)) \times \xi(u_i, r_j)], \tag{5}$$

$$t^{(n+1)}(s_i) = \frac{\sum_{r_j \in \mathcal{R}(s_i)}[f^{(n)}(r_j) \times h^{(n)}(u(r_j)) \times \rho(r_j, s_i)]}{|\mathcal{R}(s_i)|}. \tag{6}$$

In the basic model, all reviews from the same user can be considered equally reliable. However, reviews posted by the same user can have different $\xi(u_i, r_j)$, if we consider factors such as review length or the number of aspects covered in the review. This allows us to integrate detection features adopted in other approaches into the trust propagation framework. If we assume all reviews are written by users with unified confidence, $\xi(u_i, r_j)$ can be set to a constant, e.g., (1) in general, $\rho(r_j, s_i)$ measures the relevance (i.e., support) of review r_j to statement s_i, which can be computed using any sentiment analysis scheme that returns the confidence that a review expresses the same opinion as the statement does. In this work, we extract the opinions as aspect-based vectors. Therefore, $\rho(r_j, s_i)$ is set to 1 if r_j has an non-zero value for the aspect element that the statement is about, or 0 if otherwise.

It is worth noting that both models can be degenerated to the two-layer model by replacing $f^{(n)}(r_j)$ in Equation (3) (or (6)) with the corresponding form in Equation (2) (or (5)). This is because these models do not capture the influences due to the interactions on the inter-review edges.

The Enhanced Model with Inter-Review Interactions. Finally, we integrate the inter-review interactions into the enhanced model to build the complete three-layer trust propagation model that captures all types of influences due to node connectivity and interactions on these edges.

For a review r_i, based on the deviation between its opinion about the aspect category of a relevant statement s_j (denoted as $o(r_{u_i}(s_j), s_j)$) and the aggregated opinion of statement s_j (denoted as $ao(s_j)$), we define a deviation function $\Delta(o(r_i, s_j), ao(s_j))$ as

$$\Delta(o(r_i, s_j), ao(s_j)) = \begin{cases} 0, & o(r_i, s_j) = ao(s_j) \\ 1, & o(r_i, s_j) = -ao(s_j) \\ 0.5, & otherwise. \end{cases} \tag{7}$$

Corresponding, a sentiment support from a review to a statement can be defined based on the consistency between $o(r_{u_i}(s_j), s_j))$ and $ao(s_j)$. We define a support function $supp(o(r_i, s_j), ao(s_j))$ as $1 - \Delta(o(r_i, s_j), ao(s_j))$.

Here, if the sentiment polarity expressed in the review on a specific aspect category is the same as the sentiment polarity of the aggregated opinion of the statement ($o(r_i, s_j) = ao(s_j)$), then we say the review fully supports the statement. On the other hand, if the sentiment polarities between a review and a statement are totally opposite ($o(r_i, s_j) = -ao(s_j)$), i.e., positive and negative, or negative and positive, we say the review rejects the statement. For all other cases, we say the reviews partially support the statement.

The influence from the review-review edges in the inter-user context on the honesty score of a user is caused by whether the aspect category-specific opinions toward a statement (i.e., $o(r_i, s_j)$) and the corresponding aggregated opinions (i.e., $ao(s_j)$) are consistent or not. If a user's review supports the aggregated opinion and the statement has high truthfulness scores, this user will be rewarded for being consistent with a highly trusted statement. However, being inconsistent with the statement does not necessarily mean the user is dishonest. If a user's review expresses an opinion that rejects a statement with a low truthfulness score, this user should not be penalized but rewarded. The influence of an individual deviation may not always lead to the correct penalty or reward; however, based on our assumption that the benign users outnumber the spammers, the overall influence introduced by all deviations across multiple entities and aspects should reflect the changes of trust scores correctly. Therefore, we consider the overall deviation for each user, and use the average deviation in the computing:

$$\delta^{(n+1)}(u_i) = \frac{\sum_{s_j \in \mathcal{S}(u_i)} d(t^{(n)}(s_j), supp(o(r_{u_i}(s_j), s_j), ao(s_j)))}{|\mathcal{S}(u_i)|}, \tag{8}$$

where the function d models the deviation caused by the opinion difference expressed between a user's review and a statement and the trustworthiness of the statement. In particular, we defined d for the supervised-learning-based approach in Equations (9) and (10) for the unsupervised-learning-based approach. $2x - 1$ in both equations maps the score $t(s_j)$ from [0, 1] to [−1, 1].

$$d(x, y) = -y * ln\left(\frac{e^{2(2x-1)}}{1 + e^{2(2x-1)}}\right) - (1 - y) * ln\left(\frac{1}{1 + e^{2(2x-1)}}\right), \tag{9}$$

$$d(x, y) = -y * ln\left(\frac{e^{2x-1}}{1 + e^{2x-1}}\right) - (1 - y) * ln\left(\frac{1}{1 + e^{2x-1}}\right). \tag{10}$$

The only difference between Equations (9) and (10) is the coefficient used before $(2x - 1)$, which acts like an amplifier for the score $t(s_j)$. We can see that with more coarse-grained opinions, adding an amplifier makes the deviation achieve similar results as in an unsupervised setting. Based on experimental exploration, we set the coefficient as 2.

We define the function d in this form so that the deviation decreases when the trust score $t(s_j)$ is small, even when the support $supp(o(r_{u_i}(s_j), s_j), ao(s_j))$ is 0. When $supp(o(r_{u_i}(s_j), s_j), ao(s_j))$ is 1, the deviation decreases when the score $t(s_j)$ is large. On the other hand, when $supp(o(r_{u_i}(s_j), s_j), ao(s_j))$ is 0.5, the deviation increases in both directions as $t(s_j)$ increases, but at a slower speed. Finally, we compute the honesty scores for the users to reflect the influence of the overall deviation as below:

$$h^{(n+1)}(u_i) = \frac{\beta + 1}{\beta + e^{\delta^{(n+1)}(u_i)}}. \tag{11}$$

The parameter β here controls the extent the score of a user u_i is affected by his/her deviation $\delta(u_i)$. With smaller value of β, the score drops quicker as $\delta(u_i)$ increases. The measurement of trust is propagated along the structural connections. For example, a user's honesty score is dependent on the trustworthiness of the statements in his reviews, thus the trust is propagated from his statements to the user himself, and further propagates to his reviews and back to his statements. Each type of score gets feedbacks from the other two, which allows reinforcement based on the connections among the nodes.

4.3 The Computational Framework

The scores of users, reviews, and statements are computed in an iterative computational framework, as shown in Algorithm 1. In the beginning, the nodes of users, reviews, and statements are generated from reviews data. The textual content of reviews is processed for extracting the aspect category-based opinions. In each round, all nodes update their scores accordingly and do normalization after all scores are updated. After the model converges, the final result is output as the measurement of modeled trust for users, reviews, and statements.

For the convergence of the reinforcement-based iterative model, the two-layer models like HITS [28] and PageRank [53] are proved to be converged using eigenvectors in the original papers. As for three layers, to the best of our knowledge, there is no mathematical proof in models with similar structure [70, 81] as ours, since the three-layer models cannot be rewritten in matrix form. During experiments, we observe that our three-layer model always converges. With opinions extracted in the supervised setting, the model empirically converges in around 100 iterations. With unsupervised setting, the model empirically converges in around 30 iterations. We infer the reason that the model with opinions extracted from unsupervised methods converges faster is that with finer-grained opinions, the model is able to aggregate more information to compute the trust scores, and the changes of the propagated trust between two consecutive rounds are larger than the ones in the supervised approach.

ALGORITHM 1: Iterative framework to compute trust-related scores

```
Input:
Collections of users $\mathcal{U}$, reviews $\mathcal{R}$, and statements $\mathcal{S}$
Initial sentiment polarities for all statements in $\mathcal{S}$
Parameters $\mu$, $\beta$
Output:
Honesty scores $h(u)$ for all users in $\mathcal{U}$
Faithfulness scores $f(r)$ for all reviews in $\mathcal{R}$
Truthfulness scores $t(s)$ for all statements in $\mathcal{S}$
repeat
    Compute the faithfulness scores for all reviews using Equation (5)
    Compute the truthfulness scores for all statements using Equation (6)
    Compute the honesty scores for all users using Equation (11)
    Normalize each type of score with the largest as 1.0
until converged;
```

5 EXPERIMENTS AND EVALUATION

We conducted several experiments on three different datasets of the Yelp reviews. In this section, we will explain the design of the experiments and evaluate the performance of our proposed models with experiment results and a case study.

5.1 Dataset

We used three datasets in the experiments. The first one is the SemEval-2014 dataset [58] published in the Semantic Evaluation series, the second is a dataset that we crawled from Yelp.com in 2013, and the last one was shared by the authors of [48].

The SemEval dataset contains 3,041 sentences from restaurant reviews, which we used to train our classifier. In SemEval, each sentence is labeled with one or multiple aspect categories (i.e., food, service, price, ambience, and anecdotes/miscellaneous) and the corresponding sentiment polarities (i.e., positive, neutral, negative, and conflict). As discussed in Section 3, the "conflict" sentiment category is not considered in our model. We then split this dataset into a 4:1 ratio with a training dataset and a testing dataset of 2,432 and 609 labeled sentences, respectively.

The Yelp dataset that we have crawled from Yelp.com in 2013 contains 9,314,945 reviews about 125,815 restaurants in 12 U.S. cities, which were input by 1,246,453 users between 2004 and 2013. We extracted the data for the city of Palo Alto, California to test our content-aware trust propagation models. It contains 128,361 reviews about 1,144 restaurants from 45,180 users. To build the graph of our three-layer model, we conducted data cleaning to discard reviews that do not

Table 2. The Performance of Aspect Category Classification

Label	Precision	Recall	F1-score	Support	Accuracy
food	0.81	0.78	0.80	238	0.844
not_food	0.86	0.88	0.87	371	
avg/total	0.84	0.84	0.84	609	
price	0.91	0.62	0.73	65	0.952
not_price	0.96	0.99	0.97	544	
avg/total	0.95	0.95	0.95	609	
service	0.82	0.69	0.75	122	0.906
not_service	0.92	0.96	0.94	487	
avg/total	0.90	0.91	0.90	609	
ambience	0.83	0.52	0.64	84	0.920
not_ambience	0.93	0.98	0.95	525	
avg/total	0.91	0.92	0.91	609	
anecdotes/miscellaneous	0.77	0.70	0.73	243	0.796
not_anecdotes/miscellaneous	0.81	0.86	0.84	366	
avg/total	0.79	0.80	0.79	609	

contain aspect-specific opinion indicators. We also ignored the statements with less than three related reviews and the users with less than three expressed statements, and then adjusted the corresponding relationships. After cleaning, the dataset used in the supervised approach contains 46,652 reviews from 2,184 users for 1,071 restaurants. The dataset used in the unsupervised approach contains 40,064 reviews written by 2,182 users for 1,034 restaurants. Although our datasets contain rich information about the reviewers, such as the total number of reviews, average ratings, social relationships, and so on, we only used the review content in this study.

The Yelp dataset with flagged reviews collected by Mulherjee et al. in 2013 [48] is also used to evaluate the performance of our trust propagation model. It contains 788,471 reviews about 250,078 restaurants from 35,430 users.

5.2 Supervised Opinion Extraction

We first built a SVM classifier for opinion extraction. For feature extractions, we used bag-of-words and extracted the tf-idf weights as features. The classifiers for aspect categories and sentiment polarities were trained separately at the sentence level. A single sentence may contain multiple aspect categories. Since SVM is a binary classifier, a trained SVM classifier only classifies a sentence into one category, but cannot determine if it contains multiple categories. So, we trained five binary one-vs.-all SVM classifiers independently, one for each aspect category, as suggested by [26]. Once we had the classified aspect categories, we applied the trained classifiers of sentiment on each category to obtain the category-based sentiment polarities. As a result, one review may contain opinions about several aspect categories, such as "food,positive," "price,neutral," and "service,negative," which are called *aspect-specific opinions*. Therefore, the extracted opinion of a review consists of a set of aspect-specific opinions.

Extracted Aspect Categories. The results of aspect category classification are shown in Table 2, in which "avg" means the average of precision, recall, and f1-score, and "total" denotes the total support of each category.

Among the five categories, the "anecdotes/miscellaneous" category has the worst precisions and recalls, which is reasonable, since this category contains all aspects that cannot be classified

Table 3. The Classification Performance of Category-Based Sentiment Polarities

Label	Precision	Recall	F1-score	Support	Accuracy
food,negative	0.39	0.36	0.38	33	0.740
food,neutral	0.50	0.04	0.07	25	
food,positive	0.80	0.92	0.85	169	
avg/total	0.71	0.74	0.70	227	
price,negative	0.55	0.44	0.49	25	0.635
price,neutral	0.00	0.00	0.00	2	
price,positive	0.67	0.81	0.73	36	
avg/total	0.60	0.63	0.61	63	
service,negative	0.66	0.69	0.67	48	0.698
service,neutral	0.00	0.00	0.00	7	
service,positive	0.73	0.79	0.76	61	
avg/total	0.66	0.70	0.68	116	
ambience,negative	0.64	0.30	0.41	23	0.675
ambience,neutral	0.00	0.00	0.00	5	
ambience,positive	0.68	0.92	0.78	49	
avg/total	0.62	0.68	0.62	77	
anecdotes/miscellaneous,negative	0.11	0.10	0.10	31	0.547
anecdotes/miscellaneous,neutral	0.60	0.49	0.54	96	
anecdotes/miscellaneous,positive	0.60	0.73	0.66	107	
avg/total	0.54	0.55	0.54	234	

into any other category. It is easier to determine what does not belong to anecdotes/miscellaneous than to determine what does. As a result, the precision, recall, and f1-scores of not_anecdotes/miscellaneous are higher than anecdotes/miscellaneous itself. The "food" category is the most popular aspect category in restaurant reviews; interestingly, it has the second worst performance among the five categories. This is partially because there are too many different terms and aspects representing food types. Using the tf-idf weights as features, it is difficult to have a unified representation of the category. So, it is more difficult to train an effective classifier for the food category than for the price or service categories.

Extracted Aspect-Specific Opinions. We present the results of sentiment classification in Table 3. It shows only the performance of using SVM as the classifier for opinion mining, which is not the evaluation of the performance of opinion mining on the Yelp dataset. Comparing to the performance of aspect category classification, the performance of category-based sentiment polarity classification is worse. This may be because bag-of-words captures representative features for categories better than capturing sentiment polarities. Sometimes, the sentiment polarities are implicit and context-dependent. Moreover, since the category-based sentiment analysis takes the classification results for aspect categories as the input, mistakes in the previous classification will be amplified and affect the overall performance. It is worth noting that, despite all these issues, our classification performance of sentiment polarity is still better than or comparable to the baseline and some approaches in the SemEval 14 contest [58].

5.3 Unsupervised Opinion Extraction

The trust propagation model takes the classification results of aspect categories and category-based sentiment polarities as input. As discussed in Section 5.2, SVM does not yield the best results. So,

Table 4. The Aspect Categories Extracted From Aspect Clustering

1	visit, favorite, experience, star, rating, review, trip
2	drink, dish, food, juice, entree, appetizer, meal
3	veggie, rice, fish, wrap, roll, steak, burger
4	date, evening, night, event, party, occasion, birthday
5	yogurt, cream, dessert, cupcake, ice-cream, cake, pie
6	bartender, waitress, waiter, server, table, complaint, job
7	spot, patio, location, space, area, room, parking
8	choice, selection, size, amount, variety, type, combination
9	music, decor, service, ambiance, atmosphere, vibe, interior

we examine the unsupervised classification models to improve the overall performance of our model.

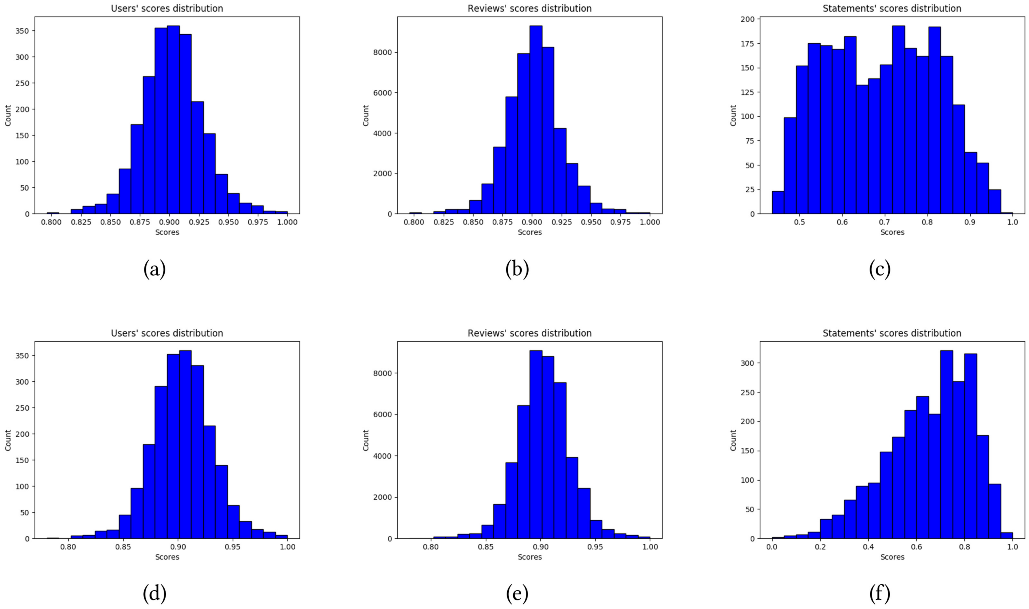
In unsupervised opinion extraction, we first parsed the reviews using the Stanford PCFG parser [27] to capture the dependency relations in parsing. Then, we computed the word embeddings to include the semantic meanings of words. In particular, we used Word2Vec [42] with the skip-gram loss function to train our word embedding model, and used the Python library gensim [63] to implement it. We chose skip-gram instead of CBoW since it performs better in semantic tasks [42]. Finally, we used K-means for aspect clustering and set the cluster number to 9 in the following experiments. Among a few values we tried for the cluster number, it generated the most reasonable results on our data.

Extracted Aspect Categories. In Table 4, we show the top-7 words of the nine clusters identified in the unsupervised aspect extraction approach. From the table, we can see that the word2vec model captures the semantic relationships between words and clusters them into meaningful aspect categories. For example, the top-7 words in cluster 7 are spot, patio, location, space, area, and so on, which represent a category about the "environment." Similarly, words in cluster 9 such as music, ambiance, atmosphere, interior, and so on, represent a semantic category of "ambience." As a result, we used these most frequent words of each cluster to represent an aspect category.

5.4 Trust Propagation with Supervised Opinion Extraction

In supervised opinion extraction, we classified the reviews into five aspect categories. However, in the experiments, we only used four categories in trust propagation, while the miscellaneous category is ignored. The miscellaneous category contains a mix of opinions on multiple aspects that cannot be classified into the other four categories. Therefore, it is often the case that one user's opinion classified as "miscellaneous" is about an entirely different aspect from another user's opinion on "miscellaneous." As a result, this category cannot contribute much to trust measurement. However, it is necessary to keep the miscellaneous category in opinion extraction so as to improve the precision of the other four categories.

In the experiments, we adopted two initialization settings for the statements. The three-layer model is constructed based on the structural relationships among users, reviews, and statements. A statement is an aspect-specific opinion expressed in the review of a user about a restaurant. For example, user u_1 writes a review that a restaurant provides good services, from which we can extract a "service, positive" statement, while user u_2 feels oppositely so his review expresses a "service, negative" statement. Obviously, with three sentiment polarities, there could be three statements for each restaurant on each aspect category. To reduce the complexity, we keep at most one statement for each restaurant on each aspect category in the framework and remove the other two.



(a) (b) (c)

(d) (e) (f)

Fig. 2. Distributions under the aggregate opinion setting: (a) the distribution of honesty score for users; (b) the distribution of faithfulness score for reviews; (c) the distribution of truthfulness score for statements. Distributions under the positive opinion setting: (d) the distribution of honesty score for users; (e) the distribution of faithfulness score for reviews; (f) the distribution of truthfulness score for statements.

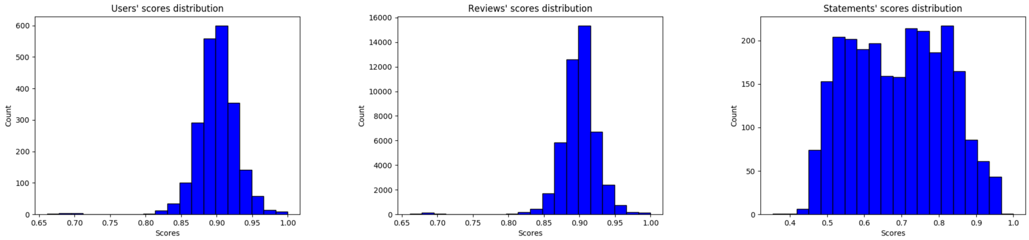
This could be done by considering the support from a review to a statement. For example, if the "service, positive" statement is selected, the support from a review expressing "service, negative" will be set to 0. To determine which statements remain in the trust propagation framework, we considered two different settings in the experiments. In the first setting, we kept only the statements that express the aggregate opinions (i.e., the opinions shared by the majority of the users) for each aspect category. In the second setting, we initiate all the statements with the positive polarity, which means all the positive statements are kept in the framework. Finally, for μ and β, we set their values to 0.5 and 1.0, respectively.

Trust Scores. We calculated the three types of trust scores under two initialization settings of the statements. First we studied the distributions of trust scores under two settings. As shown in Figure 2(a) and (b), both the honesty scores for users and the faithfulness scores for reviews follow the normal distribution with the mean around 0.75. This implies that some users may be biased or dishonest, but most of the users and their reviews are trustworthy. The distribution of the truthfulness scores for statements is shown in Figure 2(c), which is somehow skewed and pushed toward 1. This means most of the claims are highly truthful, which is reasonable since they are initialized under the aggregate opinion setting, representing the opinions of the majority.

On the contrary, when we initialize all the polarities of the statements as positive, we intend to include some false statements in the framework. Intuitively, they should receive much lower support from truthful users, and they are expected to have low truthfulness scores. This is proved by the experiment as shown in Figure 2(f). Comparing to Figure 2(c), quite a few statements have truthfulness scores lower than 0.5, while the statements with positive aggregate opinions still receive high truthfulness scores. It is worth noting that the changes in statement trust scores do

Table 5. The Average Truthfulness Scores of the Statements Under Two Initialization Settings

Category	Initialized with aggregate opinions	Initialized with all positive opinions
Food	0.771	0.796
Price	0.575	0.501
Service	0.600	0.588
Ambience	0.684	0.703



(a) (b) (c)

Fig. 3. Trust score distributions with synthetic data: (a) the distribution of honesty score for users; (b) the distribution of faithfulness score for reviews; (c) the distribution of truthfulness score for statements.

not mean our model is sensitive to initialization. It simply shows that our trust propagation model is effective in capturing and penalizing the untruthful statements by giving low scores to them.

From Figure 2(d) and (e), we can see that the honesty and faithfulness scores still demonstrate the normal distribution under the second setting. Comparing to Figure 2(a) and (b), the absolute scores of individual users and reviews change slightly, since users are affected by the false statements at different levels. However, the shape of the distribution and the relative ranking do not change much, which indicates our model is still robust to initialization settings.

Then, we compared the average truthfulness scores of four aspect categories to understand the influence of initialization settings. As shown in Table 5, in both settings, the statements about the food category received the highest average scores. Among the four categories, the scores of the "price" and "service" categories under the second initiation setting are smaller than the ones under the first setting. We observed quite a few positive statements about these two categories receiving lower trustfulness than positive statements about the other categories, which indicates the opinions on "price" and "service" are more controversial and subjective. In the second setting, these categories are more likely to be penalized more.

Evaluations. To evaluate the performance of our model, we did experiments with synthetic data and human evaluators under the first initialization setting. We first created a small synthetic dataset of 20 users, who were randomly chosen from our dataset. We manually changed their reviews so that 10 users' reviews fully support the statements relevant to their reviews, and the other 10 users' reviews fully reject all the statements. With this synthetic dataset, we aim to see if the model correctly rewards the "truthful" users who agree with the majority of the others and penalizes "untruthful" users who always deviate from the majority opinions.

From Figure 3(a) and (b), we can see two obvious clusters based on the trust scores for users and reviews. As expected, the 10 users who fully rejected the statements obtained smaller trust scores than others. Meanwhile, comparing to Figure 2(a), the scores of the remaining users, including the other 10 users with synthetic data and the users from the original dataset, increased obviously. This is a side effect introduced by the 10 users who fully supported the statements. This indicates

Table 6. The Average Honesty Scores Using Supervised Opinion Extraction and Synthetic Spamming Data

Synthetic type	Min	Average	Median	Max
Support	0.916	0.947	0.950	0.985
Reject	0.661	0.691	0.692	0.718

Table 7. The Agreements Between Our Model and Human Evaluators

	Our model	Evaluator 1	Evaluator 2	Evaluator 3
Our model	–	12	10	10
Evaluator 1	–	–	10	10
Evaluator 2	–	–	–	12

when there are more truthful users in the system, the model could better distinguish the truthful users from the spammers. We further show the average as well as the maximum and minimum honesty scores of the two synthetic groups in Table 6, which demonstrates the distinction between the two groups as expected.

Next, we recruited three human evaluators to test the performance of our model. To generate the evaluation dataset, we randomly selected 20 users from our dataset and randomly selected 8 reviews for each user. In each test, we presented two users and their reviews together as a test set to the evaluators. In each evaluation, we gave the evaluators 20 tests, which were randomly selected from a total of 190 test sets. The human evaluators were asked to read all the reviews of the two users and rank them based on their relative honesty. Then, we compared evaluators' judgments with the user honesty scores calculated by our model. We considered a judgment agrees with another if both ranked the two users in the same way. Finally, we measured the agreements between every two evaluators and between each evaluator and our model, and presented the results of one evaluation in Table 7.

Overall, the agreement between our model and the human evaluators is low, with an average of 53.3%. This is because the evaluators had difficulties in telling if one user is more honest than the other only from the 16 reviews. As we can see, the agreements between human evaluators are almost the same, which indicates the evaluators did not reach a consistent judgment among themselves. This is not very surprising, since we have observed that the human judgment is not reliable, when the content of the reviews is carefully crafted to avoid obvious content-based spamming cues. In fact, the reviews used in the evaluation have already been filtered by the Yelp Filter, which is a proprietary filter developed by Yelp.com [19, 20]. While Yelp does not reveal its filtering mechanism, some study [48] shows that the Yelp filter is likely to use behavioral features such as review length, rating deviation, percentage of positive reviews, and so on, to filter out suspicious reviews. Therefore, the reviews we presented to human evaluators had few behavioral features to help them in the judgment. This is also the reason that we propose to develop the model based on the deviation in aspect-specific opinions instead of the content-based features used in previous approaches.

5.5 Trust Propagation with Unsupervised Opinion Extraction

In this experiment, the opinions extracted from the unsupervised-based approach were input into the trust propagation model to calculate the opinion vectors and quality vectors. We used the nine aspect categories as shown in Table 4, and set the values of μ and β to 0.5 and 1.0, respectively.

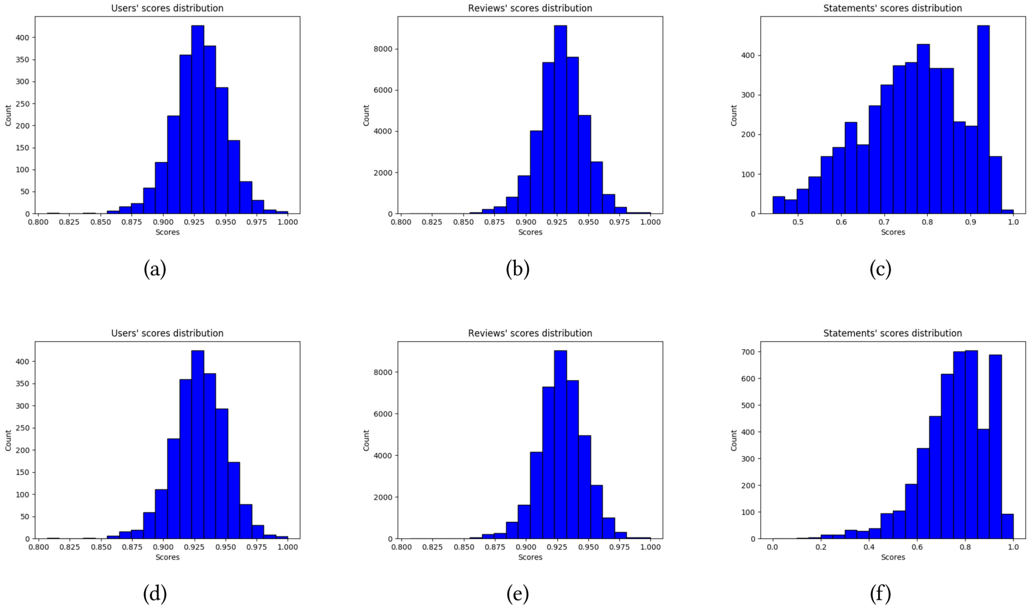


Fig. 4. Distributions under the aggregate opinion setting: (a) the distribution of honesty scores for users; (b) the distribution of faithfulness scores for reviews; (c) the distribution of truthfulness scores for statements. Distributions under the positive opinion setting: (d) the distribution of honesty scores for users; (e) the distribution of faithfulness scores for reviews; (f) the distribution of truthfulness scores for statements.

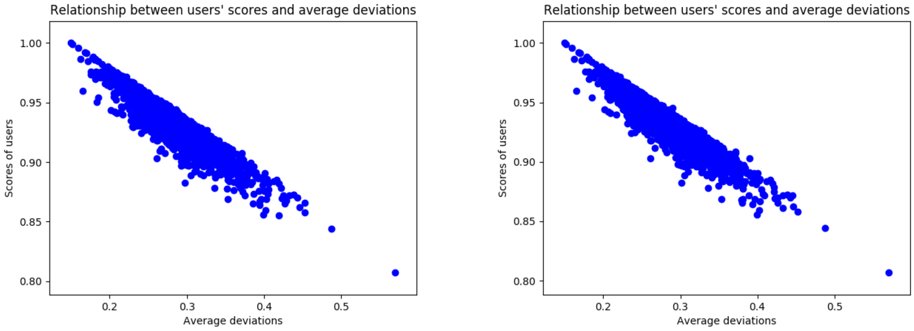
Similar to the experiment in Section 5.4, we also adopted two initialization settings, one with statements of the majority opinions and the other with statements of positive opinions.

Trust Scores. We computed the trust scores for users, reviews, and statements under two settings. Similar to the experiments with supervised opinion extraction, the honesty scores and faithfulness scores under two settings tend to be normally distributed, as shown in Figure 4(a), (b), (d), and (e). Comparing to Figure 2, the mean honesty and faithfulness scores with unsupervised opinion extraction are larger than the ones with supervised opinion extraction. We further draw the scatter plots to illustrate the relationship between a user's honesty score and the average support of her reviews to the corresponding statements. As shown in Figure 5, under both initialization settings, the honesty scores are linearly decreasing along with the increase of the average deviation.

Evaluations. Similarly to the evaluations of the model with supervised opinion extraction, we conducted two evaluations, one with synthetic data and the other with human evaluators.

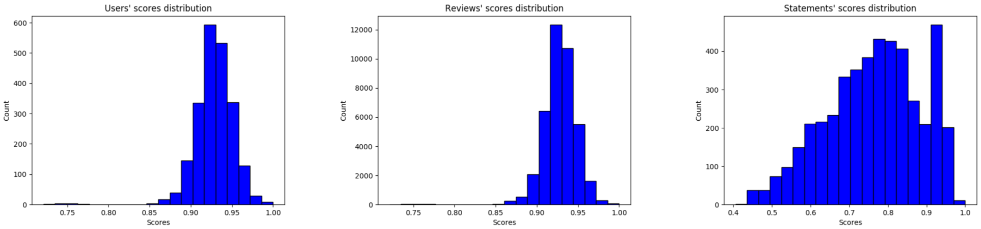
We took the same process as described in Section 5.4 to prepare the synthetic data. The distributions of the honesty, faithfulness, and trustfulness scores for users, reviews, and statements are shown in Figure 6. The honesty and faithfulness scores also demonstrated the cluster effect—the 10 users who fully rejected the statements obtained clearly lower scores than others. Table 8 shows the average, median, maximum, and minimum scores of the two groups of users with synthetic data. From the table, we see that all the users supporting the aggregate opinions obtained very high honesty scores, while all the users rejecting the aggregate opinions were penalized a lot. The results show a similar tendency as the results with supervised opinion extraction.

Five human evaluators were recruited to evaluate the model with unsupervised opinion extraction under the setting with aggregate opinions. We provided a short training to the human



(a) (b)

Fig. 5. The relationship between the honesty scores and the average deviations under the initialization setting with (a) the aggregate opinions and (b) the positive opinions.



(a) (b) (c)

Fig. 6. Trust score distributions with synthetic data: (a) the distribution of honesty scores for users; (b) the distribution of faithfulness scores for reviews; (c) the distribution of truthfulness scores for statements.

Table 8. The Average Honesty Scores Using Unsupervised Opinion Extraction with Synthetic Data

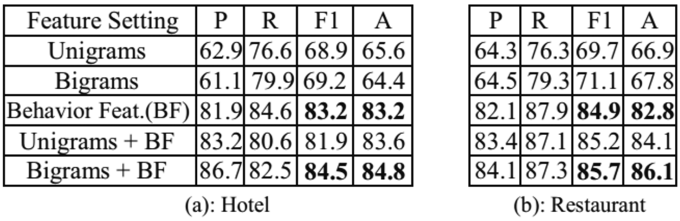
Synthetic type	Min	Average	Median	Max
Support	0.940	0.966	0.965	0.982
Reject	0.733	0.751	0.749	0.780

evaluators with information about how to determine whether a review is trustful from its content. Then, we randomly selected 10 restaurants from out dataset, and randomly selected four reviews for each restaurant. In particular, the four reviews included three reviews with high faithfulness scores and one review with a low faithfulness score. In each test, we presented one restaurant and its four reviews to the evaluator, and asked them to select the one that appeared to be the most suspicious review to them. The four reviews were displayed in a random order to eliminate the correlation between the review score and the display order.

Each evaluator completed 10 tests and thus selected 10 most suspicious reviews with her best judgments. We measured the agreements between our model and each evaluator, as well as the agreements between every two evaluators. As shown in Table 9, our model is consistent with the judgments of human evaluators and achieves an overall agreement ratio of 72%. Meanwhile, the evaluators agreed with each other in deciding the most suspicious review for each restaurant, and achieved an overall agreement of 69%.

Table 9. The Agreements Between our Model and the Evaluators.

	Our model	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4	Evaluator 5
Our model	–	8	7	9	5	7
Evaluator 1	–	–	9	8	6	8
Evaluator 2	–	–	–	7	6	7
Evaluator 3	–	–	–	–	5	7
Evaluator 4	–	–	–	–	–	6



Feature Setting	P	R	F1	A
Unigrams	62.9	76.6	68.9	65.6
Bigrams	61.1	79.9	69.2	64.4
Behavior Feat.(BF)	81.9	84.6	**83.2**	**83.2**
Unigrams + BF	83.2	80.6	81.9	83.6
Bigrams + BF	86.7	82.5	**84.5**	**84.8**

(a): Hotel

P	R	F1	A
64.3	76.3	69.7	66.9
64.5	79.3	71.1	67.8
82.1	87.9	**84.9**	**82.8**
83.4	87.1	85.2	84.1
84.1	87.3	**85.7**	**86.1**

(b): Restaurant

Fig. 7. The performance of the behavior-based detector in [48].

5.6 Evaluation on the Dataset with Yelp Flagged Reviews

To further evaluate our model, we performed experiments on a separate dataset with flagged Yelp reviews, which was collected and published by the authors of [48].

Dataset. This dataset contains 788,471 reviews about 250,078 restaurants from 35,430 users. Among the 788,471 reviews, 8,303 reviews were "flagged" by the Yelp filter [20]. These reviews were written by 7,118 users about 98 restaurants in Chicago, Illinois. Therefore, to build the dataset for evaluation, we used the 8,303 reviews as the "seeds" to extract all the restaurants reviewed by the 7,118 users as well as all the reviews and reviewers of these restaurants. The final evaluation dataset contains 100,290 reviews about 16,782 restaurants, which were submitted by 34,785 users.

The Behavior-Based Detector [48]. Mukherjee et al. proposed a detection approach that combined n-gram with behavioral features such as *maximum number of reviews*, *percentage of positive reviews*, *review length*, *reviewer rating deviation*, and *maximum content similarity* to identify spamming reviews [48]. They used the aforementioned dataset, and treated the "flagged" reviews as the "ground truth" for spamming reviews. Their results are shown in Figure 7(b), where P, R, F1, and A denote Precision, Recall, F1-score, and Accuracy, respectively. According to their results, bigrams combined with behavior features achieved the highest accuracy and F1 score of 85.7% and 86.1%, respectively.

This approach and several others following this direction treat spam detection as a classification problem, in which the performance of the classifiers highly depends on the quality of the training data. However, in our study of the flagged reviews, which were labeled or detected as spam in [48], we find several reviews are currently "unflagged" by the Yelp filter and publicly visible. These reviews receive medium to high ranks in our system, while they were marked as spam in [48]. In fact, reviews flagged by the Yelp filter may not only include spam reviews [20]. Moreover, from the above behavior features, we can see that the detector in [48] could be evaded by sophisticated spammers who write lengthy reviews, and avoid simple duplication and extreme rating behaviors. If such advanced spam reviews exist in the dataset, we are not sure how many of them were correctly flagged by the Yelp filter in 2013. Therefore, we think there still lacks an objective ground

Table 10. The Average Flagged Ratios and Deviation Ratios of Three Suspicious Groups of Users

	#Users	Avg. Flagged Ratio	Avg. Deviation Ratio	Bottom-100 Ratio
Suspicious group 1	8	1.000	1.000	0.750
Suspicious group 2	18	0.525	0.900	0.333
Suspicious group 3	59	0.205	0.534	0.169

truth to directly compare the results of our model with others'. Hence, we evaluated our model with a combined dataset of real-world data and synthetic data, as described in Sections 5.4 and 5.5.

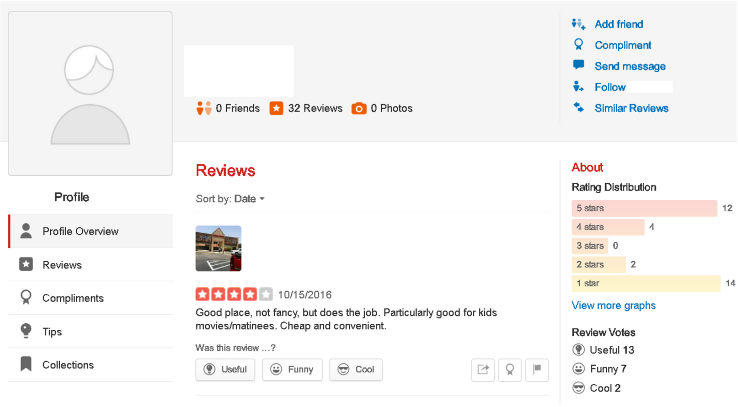
Evaluation Metrics. As explained above, the flagged reviews cannot be used directly as the ground truth. So, we select a smaller set of more suspicious users with a higher confidence to be the ground truth for evaluation. For each flagged review in the evaluation, we calculate the *flagged ratio* of its reviewer as the percentage of flagged reviews among all his/her reviews. This ratio reflects how suspicious a user is from the view of the Yelp filter. Then, we divide the users who have flagged reviews into three suspicious groups according to their flagged ratios: users with the ratio equal to 1 are in "suspicious group 1" and users with the ratio larger than 0.5 and smaller than 0.5 are put into groups 2 and 3, respectively. Obviously, users in group 1 are more suspicious than those in group 2, and the users in group 3 are the least suspicious.

Our model measures the degrees of trustworthiness of the reviews, users, and the aspect-specific review statements with three types of trust scores. So, we define two more metrics to evaluate users' honesty scores calculated by our model. First, we define the *deviation ratio* for the user as the percentage of reviews with aspect-specific deviations among all his/her reviews. This ratio reflects how suspicious a user is from the view of our trust propagation model. Then, we rank the users based on their honesty scores in descending order and study the bottom-100 set, which contains the most suspicious spammers identified by our model. We define the *bottom-100 ratio* as the percentage of users in a suspicious group who fall into the bottom-100 set.

Evaluation Results. In this evaluation, we want to study the users who are highly likely to be the spammer. So, we target the set of users whose reviews were flagged by the Yelp filter. However, besides spamming, the reviews may be flagged due to other illegitimate content. Also, some may not contain aspect-specific opinions, for example, empty or very short reviews with 5-star ratings. Therefore, we identified a small set of users whose flagged reviews are meaningful, reflecting opinions on one or multiple aspects. In particular, we extracted the aspects[1] from the reviews and filtered out the ones whose content does not cover any aspect. We further compared the opinions in each review with the aggregated opinions, and keep the ones deviating from the aggregated opinions. This reduced the set to 90 flagged reviews from 85 users. These users are more suspicious than others in the evaluation dataset viewed by both the Yelp filter and our model. Then, we calculated the flagged ratio to determine the three suspicious groups, and calculate the deviation ratio for each user and the bottom-100 ratio for each group. As shown in Table 10, 75% of the users in suspicious group 1 fall into the bottom-100 set, while only 16% of the users in suspicious group 3 are ranked to the bottom 100 by our model. This indicates that users with more flagged reviews are more likely to receive lower honesty scores from our model.

Case Study. We further investigated the other users who were ranked to the bottom-100 by our model to see if they are suspicious spammers and how suspicious they are.

[1] We conduct experiments using both supervised and unsupervised approaches, but only present the results in the unsupervised approach here since the results in both approaches yield the similar tendency.



[redacted]

0 Friends 32 Reviews 0 Photos

10/15/2016

Good place, not fancy, but does the job. Particularly good for kids movies/matinees. Cheap and convenient.

Was this review ...?

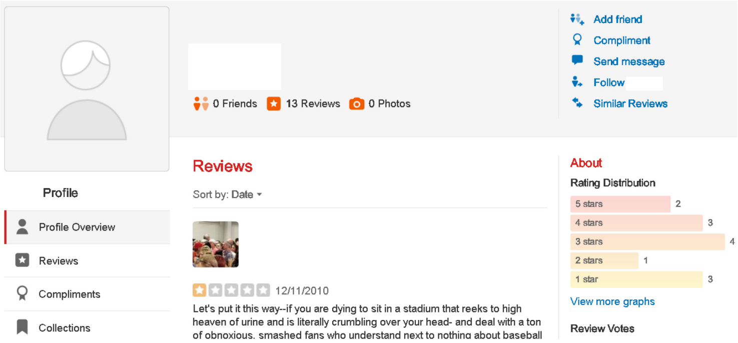
Rating Distribution

5 stars	12
4 stars	4
3 stars	0
2 stars	2
1 star	14

Review Votes

Useful 13
Funny 7
Cool 2

Fig. 8. Case study: example weak Yelp profile 1.



[redacted]

0 Friends 13 Reviews 0 Photos

12/11/2010

Let's put it this way--if you are dying to sit in a stadium that reeks to high heaven of urine and is literally crumbling over your head- and deal with a ton of obnoxious, smashed fans who understand next to nothing about baseball

Rating Distribution

5 stars	2
4 stars	3
3 stars	4
2 stars	1
1 star	3

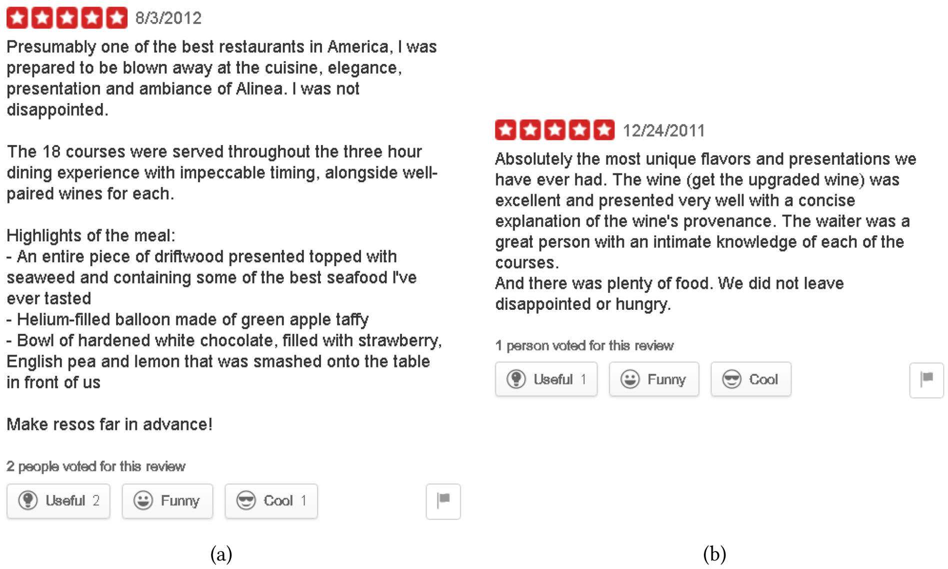
Review Votes

Fig. 9. Case study: example weak Yelp profile 2.

Since there is no ground truth, we analyzed all the information that we can find about the user, including user profiles, interactions with other users in Yelp, the reviews that are not included in the evaluation dataset, and so on; some users have weak personal profiles with little personal information. Most of these users have no friends and never interact with others. Some of them demonstrate extreme rating patterns. These are indicators of spamming accounts as recognized by other works. We show two weak user profiles[2] in Figures 8 and 9. Both profiles use the default profile picture and have no friends or photos, indicating the least effort in maintaining their online images. Moreover, the user in Figure 8 has reviewed 32 restaurants. 81.3% of his reviews have either 5-star ratings or 1-star ratings, which demonstrates extreme rating behavior.

Among the flagged reviews, we further found that several of them are now publicly visible on Yelp, which means they are no longer flagged by the Yelp filter. Figure 10 shows two examples of such reviews. Both reviews were previously flagged, but now can be found via content search in the restaurant pages. From the review content as well as the information about user profile and activities, we did not find any cue indicating spamming. This shows that not all the flagged reviews used in [48] are spam reviews and the approach of using the flagged reviews as the labeled ground truth is problematic.

[2] For privacy reasons, we removed user names, emails, and IDs of the profiles.



8/3/2012

Presumably one of the best restaurants in America, I was prepared to be blown away at the cuisine, elegance, presentation and ambiance of Alinea. I was not disappointed.

The 18 courses were served throughout the three hour dining experience with impeccable timing, alongside well-paired wines for each.

Highlights of the meal:
- An entire piece of driftwood presented topped with seaweed and containing some of the best seafood I've ever tasted
- Helium-filled balloon made of green apple taffy
- Bowl of hardened white chocolate, filled with strawberry, English pea and lemon that was smashed onto the table in front of us

Make resos far in advance!

2 people voted for this review

(a)

12/24/2011

Absolutely the most unique flavors and presentations we have ever had. The wine (get the upgraded wine) was excellent and presented very well with a concise explanation of the wine's provenance. The waiter was a great person with an intimate knowledge of each of the courses.
And there was plenty of food. We did not leave disappointed or hungry.

1 person voted for this review

(b)

Fig. 10. Case study: reviews previously flagged by the Yelp filter but publicly visible now.

6 RELATED WORK

6.1 Spam Review Detection

The problem of spam review detection was first studied in [22], which mainly focused on detecting duplicate spam reviews based on content similarity. Later on, various approaches have been proposed as the review spamming problem attracts more attention. One popular direction is to design machine learning methods based on behavioral patterns, such as rating behavior [11, 34], unexpectedness [23], temporal and spatial patterns [9, 31, 77], group behaviors [32, 47], inconsistency [49], and so on. These behavioral features are usually manually selected based on heuristics and observations. One drawback of behavioral-based methods is that spammers' behavior can change as they learn to disguise themselves as normal users. On the other hand, our approach focuses directly on opinions expressed in the review, which is more straightforward and does not require crafting the behavioral features.

With the development of deep neural networks, various deep learning–based methods have been proposed [32, 33, 64, 82]. These methods mainly constructed deep neural networks to learn the dense representation of the reviews. The advantage is that no deliberately crafted features are needed. However, these methods are more complex and take much more time to train.

The third type of approaches is content-based. [52] proposed a method based on linguistic features such as n-gram, while [10] focused on the syntactic rules. These works mainly utilized the language patterns of the reviews rather than the actual opinions as our model does. [51, 56] conducted sentiment analysis on the review. However, these works only analyzed the sentiment of each review, while we focused on finer-grained aspect-specific opinions. Compared to the aforementioned work, [12] is the closest to ours. They analyzed the aspect-specific profile of the products and compared the compatibility with the reviews'. Compared to our proposed method, their work only considered each review individually, while our method integrated the aspect-specific opinions into the trust propagation model and computed the credibility of users as well.

6.2 Trust Propagation

Trust and trust propagation have been extensively studied in literature. The general idea of reinforcement based on graph link information has been proved effective. HITS [28] and PageRank [53] are successful examples in link-based ranking computation. Trust propagation has been widely applied in recommender systems to make trustworthy predictions [14, 30, 39, 44]. In these works, trust is transmitted in a network of neighbors. The nodes in the networks are usually of the same type, for example, users. [72] built a heterogeneous graph to compute trustworthiness scores for users, reviews, and stores. Besides trust, various work also modeled the propagation of distrust [68, 84]. In our model, we do not model distrust explicitly. Instead, we added the penalty to the propagated trust when deviation from the majority happens. Compared to our work, these approaches only considered link-based information provided by the system but did not consider content information.

Trust not only can be determined by surrounding neighbors in a network, but also can be dependent on the context when it comes to content. [83] proposed a topic-based model to estimate the trustworthiness of users and tweets in Twitter. Compared to our model, their model evaluated trust based on topic similarities, while our approach models trust based on deviation of aspect-specific opinions from majority opinions. The work in [3] modeled the trust of users by comparing the content of social network posts to actual happened events. Similar to our model, this work models trust by deviation. However, our proposed model utilized deviations in finer-grained dimensions. [70] proposed a content-driven framework for computing trust of sources, evidence, and claims. The difference between this model and ours is that we extract more fine-grained information from content, while the model in [70] mainly used the similarities between content in general. Also, in [70], the inter-evidence similarity plays an important role to make sure that similar evidences get similar scores. However, the consensus opinions used in our model already represent such similarity, so we did not add the inter-evidence similarity. In addition, we also redefined the computational rules in the context of our problem.

6.3 Opinion Mining

Opinion mining has been used to analyze the opinions, sentiments, and attitudes expressed in a textual content toward a target entity. It typically includes work from two related areas: opinion aspect extraction and sentiment analysis. Aspect extraction aims to extract product features from opinionated text. The words that represent desired aspects are often nouns or noun phrases that can be captured using syntactic patterns. Thus, various methods that utilized dependency-based rules were proposed [18, 35–37, 59, 61, 62, 76, 79]. With advancements of deep learning, deep neural network can also achieve decent performance [60, 74]. Usually, extracted aspects are usually words that represent specific aspects. Thus, grouping them into high-level concepts is usually required for further analysis. Lexical tools like WordNet [43] are often used. In addition, topic modeling-based approaches are also very popular [4, 5, 24, 29, 69, 73], as they are able to extract and group aspects simultaneously.

On the other hand, the goal of sentiment analysis is to analyze the polarity orientation of the sentiment words toward a feature or a topic of the product. One common way is to use some sentiment tools directly, such as MPQA Subjectivity Lexicon [75] and SentiWordNet [2]. With such tools developed by other researchers, various works have been proposed [8, 15, 65]. Another common practice is to infer the polarity of target words using a small group of seed terms with known polarity [7, 21]. In addition, supervised learning algorithms are often applied in previous work [25, 54, 66], as well as deep learning–based approaches [6, 67].

In this work, we adopted two methods for aspect extraction. One is the supervised-based method with SVM. This method does not output specific aspect words, but assigns the high-level category and corresponding sentiment polarity directly. The other one is to use a dependency parser. We applied K-means clustering to cluster the aspect words together. For each aspect, we used a sentiment to determine its sentiment. Note that opinion mining is not the main focus and contribution of this work. The difference between our goal and typical opinion mining work is that we are not trying to improve the performance of extracting opinions. Instead, our purpose of applying opinion technique is to use the extracted aspects as deviation indicators for trustworthiness analysis.

7 CONCLUSION

In this work, we study the problem of inferring trustworthiness from the content of online reviews. We first apply opinion-mining techniques using both supervised learning and unsupervised learning algorithms to extract aspect-category-specific opinions expressed in the reviews. Then, we integrate the opinions to obtain opinion vectors for individual reviews and statements. Finally, we develop an iterative content-based computational model to compute honesty scores for users, reviews, and statements. According to the results, there exist differences of statement truthfulness across different categories. Our model shows that the trustworthiness of a user is closely related to the content of his/her reviews.

Our work can be easily extended to transfer the trust scores computed from a certain domain to similar domains. This can be achieved by assigning weights to the aspect categories extracted from the original domain. To transfer the trust scores to a similar domain, we can calculate the trust score for the new domain based on the similarities between the aspect categories extracted from the two domains (original and new). In addition, the review dataset we used in this work was collected in 2013. The structures and content in the dataset are static and there are no dynamic changes considered in our model. However, the reviews and qualities of restaurants tend to change with time. In order to consider the dynamic changes, we plan to add a temporal dimension in our model in the future.

REFERENCES

[1] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion fraud detection in online reviews by network effects. In *ICWSM*.

[2] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, Vol. 10. 2200–2204.

[3] Todd Bodnar, Conrad Tucker, Kenneth Hopkinson, and Sven G Bilén. 2014. Increasing the veracity of event detection on social media networks through user trust modeling. In *2014 IEEE International Conference on Big Data (Big Data'14)*. IEEE, 636–643.

[4] Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 804–812.

[5] Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *ACL (1)*. 347–358.

[6] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 69–78.

[7] Angela Fahrni and Manfred Klenner. 2008. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proceedings of the Symposium on Affective Language in Human and Machine (AISB'08)*. 60–63.

[8] Xing Fang and Justin Zhan. 2015. Sentiment analysis using product review data. *Journal of Big Data* 2, 1 (2015), 5.

[9] Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Exploiting burstiness in reviews for review spammer detection. In *ICWSM*.

[10] Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. 171–175.

[11] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. 2012. Distributional footprints of deceptive product reviews *ICWSM* 12 (2012), 98–105.

[12] Vanessa Wei Feng and Graeme Hirst. 2013. Detecting deceptive opinions with profile compatibility. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*. 338–346.

[13] Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, Vol. 9. 1–6.

[14] Peixin Gao, Hui Miao, John S. Baras, and Jennifer Golbeck. 2016. Star: Semiring trust inference for trust-aware social recommenders. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 301–308.

[15] Emitza Guzman and Walid Maalej. 2014. How do users like this feature? A fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE'14)*. IEEE, 153–162.

[16] Z. Hai, K. Chang, J. J. Kim, and C. C. Yang. 2014. Identifying features in opinion mining via intrinsic and extrinsic domain relevance. *IEEE Transactions on Knowledge and Data Engineering* 26, 3 (March 2014), 623–634.

[17] Zellig Sabbettai Harris. 1968. *Mathematical Structures of Language*. Krieger Publishing Company.

[18] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the t10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 168–177.

[19] Yelp Inc. 2009. Why Yelp Has A Review Filter. Retrieved on May 4, 2018 from https://www.yelpblog.com/2009/10/why-yelp-has-a-review-filter.

[20] Yelp Inc. 2010. Yelp's Recommendation Software Explained. Retrieved on May 4, 2018 from https://www.yelpblog.com/2010/03/yelp-review-filter-explained.

[21] Valentin Jijkoun and Katja Hofmann. 2009. Generating a non-English subjectivity lexicon: Relations that matter. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. 398–405.

[22] 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, 219–230.

[23] 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, 1549–1552.

[24] Yohan Jo and Alice H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 815–824.

[25] Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* 22, 2 (2006), 110–125.

[26] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval'14)*. 437–442.

[27] Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

[28] Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.

[29] Takuya Konishi, Taro Tezuka, Fuminori Kimura, and Akira Maeda. 2012. Estimating aspects in online reviews using topic model with 2-level learning. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1. 120–126.

[30] Wei-Po Lee and Chuan-Yuan Ma. 2016. Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks. *Knowledge-Based Systems* 106 (2016), 125–134.

[31] Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, and Jidong Shao. 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *ICWSM*. 634–637.

[32] Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee, and Jidong Shao. 2017. Bimodal distribution and co-bursting in review spam detection. In *Proceedings of the 26th International Conference on World Wide Web*. 1063–1072.

[33] Luyang Li, Bing Qin, Wenjing Ren, and Ting Liu. 2017. Document representation and feature combination for deceptive spam review detection. *Neurocomputing* 254 (2017), 33–41.

[34] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, New York, 939–948.

[35] Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2013. A logic programming approach to aspect extraction in opinion mining. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI'13) and Intelligent Agent Technologies (IAT'13)*, Vol. 1. IEEE, 276–283.

[36] Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *IJCAI*, Vol. 15. 1291–1297.

[37] Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving opinion aspect extraction using semantic similarity and aspect associations. In *AAAI*. 2986–2992.

[38] Michael Luca and Georgios Zervas. 2016. Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science* 62, 12 (2016), 3412–3427.

[39] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*. ACM, 17–24.

[40] Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining (ICDM'12)*. IEEE, 1020–1025.

[41] Igor' Aleksandrovič Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. SUNY Press.

[42] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Arxiv:1301.3781*.

[43] George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11 (1995), 39–41.

[44] Parham Moradi and Sajad Ahmadian. 2015. A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Systems with Applications* 42, 21 (2015), 7386–7398.

[45] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 632–640.

[46] Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 339–348.

[47] Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. 2011. Detecting group review spam. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW'11)*. 93–94.

[48] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S. Glance. 2013. What Yelp fake review filter might be doing? In *ICWSM*.

[49] Subhabrata Mukherjee, Sourav Dutta, and Gerhard Weikum. 2016. Credible review detection with limited information using consistency features. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 195–213.

[50] nbcnews. 2014. Be Wary of Awesome and Scathing Online Reviews. Retrieved on May 4, 2018 from https://www.nbcnews.com/business/consumer/be-wary-awesome-scathing-online-reviews-n72116.

[51] Myle Ott, Claire Cardie, and Jeffrey T. Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 497–501.

[52] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. [n. d.]. 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*.

[53] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The Pagerank Citation Ranking: Bringing Order to the Web*. Stanford InfoLab Technical Report.

[54] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 271.

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

[56] Qingxi Peng and Ming Zhong. 2014. Detecting spam review through sentiment analysis. *Journal of Software* 9, 8 (2014), 2065–2072.

[57] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.

[58] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval'14)*. 27–35.

[59] Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural Language Processing and Text Mining*. Springer, 9–28.

[60] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108 (2016), 42–49.

[61] Soujanya Poria, Erik Cambria, Lun-Wei Ku, Chen Gui, and Alexander Gelbukh. 2014. A rule-based approach to aspect extraction from product reviews. In *Proceedings of the 2nd Workshop on Natural Language Processing for Social Media (SocialNLP'14)*. 28–37.

[62] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37, 1 (2011), 9–27.

[63] Radim Řehuřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, 45–50.

[64] Yafeng Ren and Donghong Ji. 2017. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences* 385 (2017), 213–224.

[65] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. 2016. Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management* 52, 1 (2016), 5–19.

[66] Franco Salvetti, Stephen Lewis, and Christoph Reichenbach. 2004. Automatic opinion polarity classification of movie. *Colorado Research in Linguistics* 17, 1 (2004), 2.

[67] Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 959–962.

[68] Jiliang Tang, Xia Hu, and Huan Liu. 2014. Is distrust the negation of trust?: The value of distrust in social media. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*. ACM, 148–157.

[69] Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*. ACM, 111–120.

[70] V. G. Vydiswaran, Cheng Xiang Zhai, and Dan Roth. 2011. Content-driven trust propagation framework. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 974–982.

[71] Guan Wang, Sihong Xie, Bing Liu, and Philip S. Yu. [n. d.]. Review graph based online store review spammer detection. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*.

[72] Guan Wang, Sihong Xie, Bing Liu, and Philip S. Yu. 2012. Identify online store review spammers via social review graph. *ACM Transactions on Intelligent Systems and Technology* 3, 4 (Sept. 2012), Article 61, 21 pages.

[73] Tao Wang, Yi Cai, Ho-fung Leung, Raymond Y. K. Lau, Qing Li, and Huaqing Min. 2014. Product aspect extraction supervised with online domain knowledge. *Knowledge-Based Systems* 71 (2014), 86–100.

[74] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*. 3316–3322.

[75] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics* 35, 3 (2009), 399–433.

[76] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 (EMNLP'09)*. Association for Computational Linguistics, Stroudsburg, PA, 1533–1541.

[77] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu. 2012. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, 823–831.

[78] Hao Xue and Fengjun Li. 2017. A content-aware trust index for online review spam detection. In *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 489–508.

[79] Zhijun Yan, Meiming Xing, Dongsong Zhang, and Baizhang Ma. 2015. EXPRS: An extended pagerank method for product feature extraction from online consumer reviews. *Information & Management* 52, 7 (2015), 850–858.

[80] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y. Zhao. 2017. Automated crowdturfing attacks and defenses in online review systems. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1143–1158.

[81] Xiaoxin Yin, Jiawei Han, and S. Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.

[82] Zhenni You, Tieyun Qian, and Bing Liu. 2018. An attribute enhanced domain adaptive model for cold-start spam review detection. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1884–1895.

[83] Liang Zhao, Ting Hua, Chang-Tien Lu, and Ray Chen. 2016. A topic-focused trust model for Twitter. *Computer Communications* 76 (2016), 1–11.

[84] Cai-Nicolas Ziegler and Georg Lausen. 2005. Propagation models for trust and distrust in social networks. *Information Systems Frontiers* 7, 4–5 (2005), 337–358.

Received May 2018; revised December 2018; accepted January 2019