Credible Review Detection with Limited Information Using Consistency Features

Subhabrata Mukherjee(✉), Sourav Dutta, and Gerhard Weikum

Max Planck Institute for Informatics, Saarbrücken, Germany
{smukherjee,sdutta,weikum}@mpi-inf.mpg.de

Abstract. Online reviews provide viewpoints on the strengths and shortcomings of products/services, influencing potential customers' purchasing decisions. However, the proliferation of *non-credible* reviews — either fake (promoting/ demoting an item), incompetent (involving irrelevant aspects), or biased — entails the problem of identifying *credible* reviews. Prior works involve classifiers harnessing rich information about items/users — which might not be readily available in several domains — that provide only limited interpretability as to why a review is deemed non-credible.

This paper presents a novel approach to address the above issues. We utilize latent topic models leveraging review texts, item ratings, and timestamps to derive *consistency features* without relying on item/user histories, unavailable for "long-tail" items/users. We develop models, for computing review credibility scores to provide interpretable evidence for non-credible reviews, that are also transferable to other domains — addressing the scarcity of labeled data. Experiments on real-world datasets demonstrate improvements over state-of-the-art baselines.

1 Introduction

Motivation: Online reviews about hotels, restaurants, consumer goods, movies, books, drugs, etc. are an invaluable resource for Internet users, providing a wealth of related information for potential customers. Unfortunately, corresponding forums such as TripAdvisor, Yelp, Amazon, and others are being increasingly game to manipulative and deceptive reviews: fake (to promote or demote some item), incompetent (rating an item based on irrelevant aspects), or biased (giving a distorted and inconsistent view of the item). For example, recent studies depict that 20 % of Yelp reviews might be fake and Yelp internally rejects 16 % of user submissions [20] as "not-recommended".

Starting with the work of [11], research efforts have been undertaken to automatically detect non-credible reviews. In parallel, industry (e.g., stakeholders such as Yelp) has developed its own standards[1] to filter out "illegitimate" reviews. Although details are not disclosed, studies suggest that these filters tend to be fairly crude [24]; for instance, exploiting user activity like the number

[1] officialblog.yelp.com/2009/10/why-yelp-has-a-review-filter.html.

P. Frasconi et al. (Eds.): ECML PKDD 2016, Part II, LNAI 9852, pp. 195–213, 2016.
DOI: 10.1007/978-3-319-46227-1_13

of reviews posted, and treating users whose ratings show high deviation from the mean/majority ratings as suspicious. Such a policy seems to over-emphasize trusted long-term contributors and suppress outlier opinions off the mainstream. Moreover, these filters also employ several aggregated metadata, and are thus hardly viable for "long tail" items having very few reviews.

State of the Art: Existing research has cast the problem of review credibility into a binary classification task, a review is either credible or deceptive, using supervised and semi-supervised methods that largely rely on features about users and their activities as well as statistics about item ratings. Most techniques also consider spatio-temporal patterns of user activities like IP addresses or user locations (e.g., [14,15]), burstiness of posts on an item or an item group (e.g., [6]), and further correlation measures across users and items (e.g., [25]). However, the classifiers built this way are mostly geared for popular items, and the meta-information about user histories and activity correlations are not always available. For example, someone interested in opinions on a new art film or a "long-tail" bed-and-breakfast in a rarely visited town, is not helped at all by the above methods. Several existing works [21,26,27] consider the textual content of user reviews for tackling opinion spam by using word-level unigrams or bigrams as features, along with specific lexicons (e.g., LIWC [28] psycholinguistic lexicon, WordNet Affect [30]), to learn latent topic models and classifiers (e.g., [16]). Although these methods achieve high classification accuracy, they do not provide any interpretable evidence as to why a certain review is classified as non-credible.

Problem Statement: This paper focuses on detecting credible reviews *with limited information*, namely, in the absence of rich data about user histories, community-wide correlations, and for "long-tail" items. In the extreme case, we are provided with only the review texts and ratings for an item. Our goal is then to compute a *credibility score* for the reviews and to provide possibly *interpretable evidence* for explaining why certain reviews have been categorized as non-credible.

Approach: Our proposed method to this end is to learn a model based on *latent topic models* and combining them with limited metadata to provide a novel notion of *consistency features* characterizing each review. We use the LDA-based Joint Sentiment Topic model (JST) [18] to cast the user review texts into a number of informative facets — per-item, aggregating the text among all reviews for the same item, and also per-review. This allows us to identify, score, and highlight inconsistencies that may appear between a review and the community's overall characterization of an item. Additionally, we learn inconsistencies such as discrepancy between the contents of a review and its rating, and temporal "bursts" — where a number of reviews are written in a short span of time targeting an item. We propose five kinds of inconsistencies in our credibility scoring model, fed into a Support Vector Machine for classification, or for ordinal ranking.

Contribution: In summary, our contributions are summarized as:

- *Model:* We develop a novel *consistency model* for credibility analysis of reviews that works with limited information, with particular attention to "long-tail" items, and offers interpretable evidence for reviews classified as non-credible.
- *Tasks:* We investigate how credibility scores affect the overall ranking of items. To address the scarcity of labeled training data, we transfer the learned model from Yelp to Amazon to rank top-selling items based on (classified) *credible* user reviews. In the presence of proxy labels for item "goodness" (e.g., item sales rank), we develop a better ranking model for domain adaptation.
- *Experiments:* We perform extensive experiments in TripAdvisor, Yelp, and Amazon to demonstrate the viability of our method and its advantages over state-of-the-art baselines in dealing with "long-tail" items and providing interpretable evidence.

2 Related Work

Previous works on fake review/opinion spam detection focused on 2 different aspects:

Linguistic Analysis [21,26,27] – This approach exploits the distributional difference in the wordings of authentic and manually-created fake reviews using word-level features. However, such artificially created fake review datasets give away explicit features not dominant in real-world data, as confirmed by a study on Yelp filtered reviews [24], where the n-gram features performed poorly. Additionally, linguistic features such as *text sentiment* [33], *readability score* (e.g., Automated readability index (ARI), Flesch reading ease, etc.) [9], *textual coherence* [21], and rules based on *Probabilistic Context Free Grammar* (PCFG) [7] have been studied in this context.

Rating and Activity Analysis – In the absence of proper ground-truth data, prior works make simplistic assumptions, e.g., duplicates and near-duplicates are fake, and make use of *extensive* background information like brand name, item description, user history, IP addresses and location, etc. [10,11,14,17,22–24,29,32]. Thereafter, regression models trained on all these features are used to classify reviews as credible or deceptive. Some of these works also use crude or ad-hoc language features like content similarity, presence of literals, numerals, and capitalization. In contrast to these works, our approach uses limited information about users and items catering to a broad domain of applications. We harvest several consistency features from user rating and review text that give some interpretation as to why a review should be deemed non-credible.

Learning to Rank – Supervised models have also been developed to rank items from constructed item feature vectors [19]. Such techniques optimize measures like Discounted Cumulative Gain, Kendall-Tau, and Reciprocal Rank to generate item ranking similar to the training data based on the feature vectors. We use one such technique, and show its performance can be improved by removing non-credible item reviews.

3 Review Credibility Analysis

3.1 Language Model

Previous works [3,21,26,27] in linguistic analysis explore distributional difference in the wordings between deceptive and authentic reviews. In general, authentic reviews tend to have more *sensorial and concrete language* than deceptive reviews, with higher usage of nouns, adjectives, prepositions, determiners, and coordinating conjunctions; whereas deceptive reviews were shown to use more verbs, adverbs, and superlatives manifested in exaggeration for imaginary writing. [26,27] found that authentic hotel reviews are more specific about spatial configurations (small room, low ceiling, etc.) and aspects like location, amenities and cost; whereas deceptive reviews focus on aspects external to the item being reviewed (like traffic jam, children, etc.). Extreme opinions were also found to be dominant in deceptive reviews to assert stances, whereas authentic reviews have a more balanced view. Our latent facet model implicitly exploits these features to find opinion on important item facets and the overall rating distribution.

In order to explicitly capture such distributional difference in the language of credible and non-credible reviews at word-level, we use unigram and bigram language features shown to outperform other fine-grained psycholinguistic features (e.g., LIWC lexicon) and Part-of-Speech tags [27]. We also experimented with WordNet Affect to capture fine-grained emotional dimensions (like anger, hatred, and confidence), which, however, were seen not to perform well. In general, the bigram features capture context-dependent information to some extent, and together with simple unigram features performed the best, with the presence or absence of words mattering more than their frequency for credibility analysis. In our model, all the features were length normalized, retaining punctuations (like '!') and capitalization as non-credible reviews manifesting exaggeration tend to over-use the latter features (e.g., "the hotel was AWESOME !!!").

Feature vector construction: Consider a vocabulary V of unique unigrams and bigrams in the corpus (after removing stop words). For each token type $f_i \in V$ and each review d_j, we compute the presence/absence of words, w_{ij}, of type f_i occurring in d_j, thus constructing a feature vector $F^L(d_j) = \langle w_{ij} = I(w_{ij} = f_i) \;/\; length(d_j) \rangle, \forall i$, with $I(.)$ denoting an indicator function (notations used are presented in Table 1).

3.2 Facet Model

Given review snippets like "the hotel offers free wi-fi", we now aim to find the different facets present in the reviews along with their corresponding sentiment polarities by extracting the *latent* facets from the review text, without the help of any explicit facet or seed words, e.g., ideally "wi-fi" should be mapped to a latent facet cluster like "network, Internet, computer, access, ...". We also want to extract the sentiment expressed in the review about the facet. Interestingly, although "free" does not have a polarity of its own, in the above example "free" in conjunction with "wi-fi" expresses a positive sentiment of a service being offered

without charge. The hope is that although "free" does not have an individual polarity, it appears in the neighborhood of words that have known polarities (from lexicons). This helps in the joint discovery of facets and sentiment labels, as "free wi-fi" and "internet without extra charge" should ideally map to the same facet cluster with similar polarities using their co-occurrence with similar words with positive polarities. In this work, we use the Joint Sentiment Topic Model approach (JST) [18] to jointly discover the latent facets along with their expressed polarities.

Consider a set of reviews $\langle D \rangle$ written by users $\langle U \rangle$ on a set of items $\langle I \rangle$, with $r_d \in R$ being the rating assigned to review $d \in D$. Each review document d consists of a sequence of words N_d denoted by $\{w_1, w_2, ...w_{N_d}\}$, and each word is drawn from a vocabulary V indexed by $1, 2, ..V$. Consider a set of facet assignments $z = \{z_1, z_2, ...z_K\}$ and sentiment label assignments $l = \{l_1, l_2, ...l_L\}$ for d, where each z_i can be from a set of K possible facets, and each label l_i is from a set of L possible sentiment labels.

JST adds a layer of sentiment in addition to the topics as in standard LDA [1]. It assumes each document d to be associated with a multinomial distribution θ_d over facets z and sentiment labels l with a symmetric Dirichlet prior α. $\theta_d(z, l)$ denotes the probability of occurrence of facet z with polarity l in document d. Topics have a multinomial distribution $\phi_{z,l}$ over words drawn from a vocabulary V with a symmetric Dirichlet prior β. $\phi_{z,l}(w)$ denotes the probability of the word w belonging to the facet z with polarity l. In the generative process, a sentiment label l is first chosen from a document-specific rating distribution π_d with a symmetric Dirichlet prior γ. Thereafter, a facet z from θ_d conditioned on l is chosen, and subsequently a word w from ϕ conditioned on z and l. Exact inference is not possible due to intractable coupling between Θ and Φ, and thus we use Collapsed Gibbs Sampling for approximate inference.

Let $n(d, z, l, w)$ denote the count of the word w occurring in document d belonging to the facet z with polarity l. The conditional distribution for the latent variable z (with components z_1 to z_K) and l (with components l_1 to l_L) is given by:

$$P(z_i = k, l_i = j | w_i = w, z_{-i}, l_{-i}, w_{-i}) \propto$$
$$\frac{n(d,k,j,.) + \alpha}{\sum_k n(d,k,j,.) + K\alpha} \times \frac{n(.,k,j,w) + \beta}{\sum_w n(.,k,j,w) + V\beta} \times \frac{n(d,.,j,.) + \gamma}{\sum_j n(d,.,j,.) + L\gamma} \quad (1)$$

In the above equation, the operator (.) in the count indicates marginalization, i.e., summing up the counts over all values for the corresponding position in $n(d, z, l, w)$, and the subscript $-i$ denotes the value of a variable excluding the data at the i^{th} position.

3.3 Consistency Features

We extract the following features from the latent facet model enabling us to detect *inconsistencies* in reviews and ratings of items for credibility analysis.

1. **User Review – Facet Description:** The facet-label distribution of different items differ; i.e., for some items, certain facets (with polarity) are more important than others. For instance, the "battery life" and "ease of use" for consumer electronics are more important than "color"; for hotels, certain services are available for free (e.g., wi-fi) which may be charged elsewhere. Hence, user reviews involving less relevant facets of the item, e.g., downrating hotels for "not allowing pets", should also be detected.

Given a review $d(i)$ on an item $i \in I$ with a sequence of words $\{w\}$ and previously learned Φ, its facet label distribution $\Phi_d^{'}(i)$ with dimension $K \times L$ is given by:

$$\phi_{k,l}^{'} = \sum_{w:l^*=argmax_l \ \phi_{k,l}(w)} \phi_{k,l^*}(w) \tag{2}$$

For each word, w, and latent facet dimension, k, we consider the sentiment label l^* that maximizes the facet-label-word distribution $\phi_{k,l}(w)$, and is aggregated over all words. This facet-label distribution for review $\Phi_d^{'}(i)$ (dimension $K \times L$) forms a feature vector capturing the importance of various latent dimensions and *domain-specific* facet-labels.

2. **User Review — Rating:** The user-assigned rating corresponding to the review should be consistent to her opinion expressed in the review text. For example, the user is unlikely to give an average rating to an item when she expresses a positive opinion about all the important facets of the item. The inferred rating distribution $\pi_d^{'}$ (with dimension L) of a review d consisting of a sequence of words $\{w\}$ and learned Φ is computed as:

$$\pi_l^{'} = \sum_{w,k:\{k^*,l^*\}=argmax_{k,l} \ \phi_{k,l}(w)} \phi_{k^*,l^*}(w) \tag{3}$$

For each word, we consider the facet and label that jointly maximizes the facet-label-word distribution, and aggregate over all the words and facets. The absolute deviation (of dimension L) between the user-assigned rating π_d, and estimated rating $\pi_d^{'}$ from user text is taken as a component in the overall feature vector.

3. **User Rating:** Previous works [9,27,31] on opinion spam found that fake reviews tend to have overtly positive or overtly negative opinions. Therefore, we also use $\pi_d^{'}$ as a component of the overall feature vector to detect cues from such extreme ratings.

4. **Temporal Burst:** Typically observed in *group spamming*, where a number of reviews are posted in a short span of time. Consider a set of reviews $\{d_j\}$ posted at timepoints $\{t_j\}$ for a *specific* item. The burstiness of review d_i for the item is $\left(\sum_{j,j\neq i} \frac{1}{1+e^{t_i-t_j}}\right)$, with exponential decay used to weigh the temporal proximity of reviews for burst.

5. **User Review – Item Description:** In general, the description of the facets in an item review should not differ much from that of the majority. For example, if majority says the "hotel offers free wi-fi", and a user review says "internet

is charged" — this presents a possible inconsistency. For the facet model this corresponds to word clusters having the same facet label but different sentiment labels. However, experimentally we found this feature to play a weak role in the presence of other inconsistency features.

We aggregate the *per-review* facet distribution $\phi^{'}_{k,l}$ over all the reviews $d(i)$ on the item i to obtain the facet-label distribution $\Phi^{''}(i)$ of the item. We use the Jensen-Shannon divergence, a symmetric and smoothed version of the Kullback-Leibler divergence as a feature. This depicts how much the facet-label distribution in the given review diverges from the general opinion of other people about the item.

$$JSD(\Phi^{'}_{d}(i) \mid\mid \Phi^{''}(i)) = \frac{1}{2}(D(\Phi^{'}_{d}(i) \mid\mid M) + D(\Phi^{''}(i) \mid\mid M)) \tag{4}$$

where, $M = \frac{1}{2}(\Phi^{'}_{d}(i) + \Phi^{''}(i))$, and D represents Kullback-Leibler divergence.

Feature vector construction: For each review d_j, the above *consistency features* are computed, and a facet feature vector $\langle F^T(d_j)\rangle$ of dimension $2 + K \times L + 2L$ is created.

3.4 Behavioral Model

Earlier works [10,11,17] on review spam show that user-dependent models detecting user-preferences and biases perform well in credibility analysis. However, such information is not always available, especially for newcomers, and not so active users in the community. Besides, [22,23] show that spammers tend to open multiple fake accounts to write reviews for malicious activities — using each of those accounts sparsely to avoid detection. Therefore, instead of relying on extensive user history, we use simple proxies for user activity that are easier to aggregate from the community:

1. **User Posts:** number of posts written by the user in the community.
2. **Review Length:** longer reviews tend to go off-topic (high emotional digression).
3. **User Rating Behavior:** absolute deviation of the review rating from the mean and median rating of the user to other items, as well as the first three moments of the user rating distribution — capturing whether a user has a *typical rating behavior*.
4. **Item Rating Pattern:** absolute deviation of the item rating from the mean and median rating obtained from other users captures the extent to which the user disagrees with other users about the item quality; the first three moments of the item rating distribution captures the general item rating pattern.
5. **User Friends:** number of friends of the user.
6. **User Check-in:** if the user checked-in the hotel — first hand experience of the user adds to the review credibility.
7. **Elite:** elite status of the user in the community.
8. **Review helpfulness:** number of up-votes received to capture the quality of reviews.

Note that user rating behavior and item rating pattern are also captured *implicitly* using the consistency features in the latent facet model.

Since we aim to detect credible reviews in scenarios of limited information, we split the above activity or behavioral features into two components: (a) $Activity^{-}$ using features $[1-4]$, obtained straightforward from the tuple $\langle userId, itemId, review, rating\rangle$ and easily available even for "long-tail" items/users; and (b) $Activity^{+}$ using all the features. However the latter requires additional information (features $[5--8]$) that might not always be available, or takes long time to aggregate for new items/users.

Feature vector construction: For each review d_j by user u_k, we construct a behavioral feature vector $\langle F^B(d_j)\rangle$ using the above features.

3.5 Application Oriented Tasks

Credible Review Classification: In the first task, we *classify* reviews as *credible* or not. For each review d_j by user u_k, we construct the joint feature vector $F(d_j) = F^L(d_j) \cup F^T(d_j) \cup F^B(d_j)$, and use Support Vector Machines (SVM) [4] for classification of the reviews. SVM maps the features (using Kernels) to a high dimensional space, and constructs a hyperplane to separate the two categories. Although there can be an infinite number of such hyperplanes possible, SVM constructs the one with the largest functional margin given by the distance of the nearest point to the hyperplane on each side of it. New points are mapped to the same space and classified to a category based on which side of the hyperplane it lies. We use a linear kernel shown to perform the best for text classification tasks. We use the L_2 regularized L_2 loss SVM with dual formulation from the LibLinear package (csie.ntu.edu.tw/cjlin/liblinear) [5], and report 10-fold cross-validation classification accuracy on TripAdvisor and Yelp datasets.

Item Ranking: Due to the scarcity of ground-truth data pertaining to review credibility, a more suitable way to evaluate our model is to examine the *effect* of non-credible reviews on the relative *ranking* of items in the community. For instance, in case of popular items with large number of reviews, even if a fraction of it were non-credible, its effect would not be so severe as would be on "long-tail" items with fewer reviews.

A simple way to find the "goodness" of an item is to aggregate ratings of all reviews – using which we also obtain a ranking of items. We use our model to filter out non-credible reviews, aggregate ratings of credible reviews, and recompute the item ranks.

Evaluation Measures – We use the *Kendall-Tau Rank Correlation Co-efficient* (τ) to find effectiveness of the rankings, against a *reference ranking* — for instance, the *sales rank* of items in Amazon. τ measures the number of concordant and discordant pairs, to find whether the ranks of two elements agree or not based on their scores, out of the total number of combinations possible. Given a set of observations $\{x, y\}$, any pair of observations (x_i, y_i) and (x_j, y_j), where $i \neq j$, are said to be *concordant* if either $x_i > x_j$ and $y_i > y_j$, or $x_i < x_j$

and $y_i < y_j$, and *discordant* otherwise. If $x_i = x_j$ or $y_i = y_j$, the ranks are tied — neither discordant, nor concordant.

We use the *Kendall-Tau-B* measure (τ_b) which allows for rank adjustment. Consider n_c, n_d, t_x, and t_y to be the number of concordant, discordant, tied pairs on x, and tied pairs on y respectively, whereby Kendall-Tau-B is given by: $\frac{n_c - n_d}{\sqrt{(n_c+n_d+t_x)(n_c+n_d+t_y)}}$.

However, this is a conservative estimate as multiple items — typically the top-selling ones in Amazon — have the same rating. Therefore, we use a second estimate (*Kendall-Tau-M* (τ_m)) considering non-zero tied ranks to be concordant. Note that, an item can have a zero-rank if all of its reviews are classified as non-credible. A high positive (or, negative) value of Kendall-Tau indicates the two series are positively (or, negatively) correlated; whereas a value close to zero indicates they are independent.

Domain Transfer from Yelp to Amazon – A typical issue in credibility analysis task is the scarcity of labeled training data. In the first task, we use labels from the Yelp Spam Filter (considered to be the industry standard) to train our model. However, such ground-truth labels are not available in Amazon. Although, in principle, we can train a model M_{Yelp} on Yelp, and use it to filter out non-credible reviews in Amazon.

Transferring the learned model from Yelp to Amazon (or other domains) entails using the learned weights of *features* in Yelp that are analogous to the ones in Amazon. However, this process encounters the following issues:

- Facet distribution of Yelp (food and restaurants) is different from that of Amazon (products such as software, and consumer electronics). Therefore, the facet-label distribution and the corresponding learned feature weights from Yelp cannot be directly used, as the latent dimensions are different.
- Additionally, specific metadata like check-in, user-friends, and elite-status are missing in Amazon.

However, the learned weights for the following features can still be directly used:

- Certain unigrams and bigrams, especially those depicting opinion, that occur in both domains.
- Behavioral features like user and item rating patterns, review count and length, and usefulness votes.
- Deviation features derived from *Amazon-specific* facet-label distribution that is obtained using the JST model on Amazon corpus:
  - Deviation (with dimension L) of the user assigned rating from that inferred from review content.
  - Distribution (with dimension L) of positive and negative sentiment as expressed in the review.
  - Divergence, as a unary feature, of the facet-label distribution in the review from the aggregated distribution over other reviews on a given item.
  - Burstiness, as a unary feature, of the review.

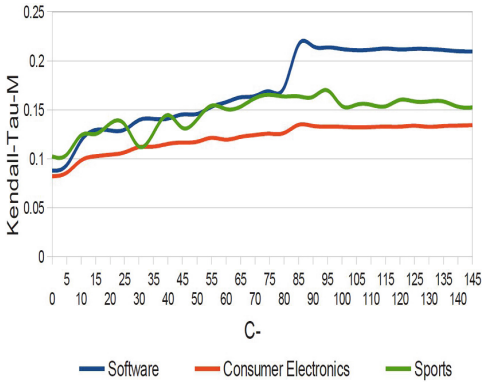

Fig. 1. Variation of Kendall-Tau-M (τ_m) on different Amazon domains with parameter C^- variation (using model M_{Yelp} trained in Yelp and tested in Amazon).

Table 1. List of variables and notations used with corresponding description.

Notation	Description
U, D, I	set of users, reviews, and items resp.
d, r_d	review text and associated rating
V, f	unigrams and bigrams vocab. &token types
w_{ij}	word of token type f_i in review d_j
$I(\cdot)$	indicator fn. for presence/absence of words
z, l	set of facets and sentiment labels resp.
K, L	cardinality of facets and sentiment labels
$\theta_d(z, l)$	multinom. prob. distr. of facet z with sentiment label l in document d
$\phi_{z,l}(w)$	multinom. prob. distr. of word w belonging to facet z with sentiment label l
Φ', Φ''	facet-label distr. of review and item resp.
α, β, γ	Dirichlet priors
π, π'	review rating distr. &inferred rating distr.
$n(\cdot)$	word count in reviews
$F^x(d_j)$	feature vec. of review d_j using lang. (x=L), consistency (x=T), and behavior (x=B)
C^+, C^-	C-SVM regularization parameters

Using the above components, that are common to both Yelp and Amazon, we *first* re-train the model M_{Yelp} from Yelp to remove the non-contributing features for Amazon.

Now, a direct transfer of the model weights from Yelp to Amazon assumes the distribution of credible to non-credible reviews, and corresponding feature importance, to be the same in both domains — which is not necessarily true. In order to boost certain features to better identify non-credible reviews in Amazon, we tune the *soft margin parameter* C in the SVM. We use *C-SVM* [2], with slack variables, that optimizes:

$$min_{\boldsymbol{w},b,\xi_i \geq 0} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C^+ \sum_{y_i=+1} \xi_i + C^- \sum_{y_i=-1} \xi_i$$
$$\text{subject to } \forall\{(\boldsymbol{x_i}, y_i)\}, y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \tag{5}$$

C^+ and C^- are regularization parameters for positive and negative class (credible and deceptive), respectively. The parameters $\{C\}$ provide a trade off as to how wide the margin can be made by moving around certain points which incurs a penalty of $\{C\xi_i\}$. A high value of C^- places a large penalty for misclassifying instances from the negative class, and therefore boosts certain features from that class. As the value of C^- increases, the model classifies more reviews as non-credible. In the worse case, all reviews of an item are deemed as non-credible, with the aggregated item rating being 0.

Table 2. Dataset statistics for review classification. (Yelp* denotes balanced dataset using random sampling.)

Dataset	Non-credible reviews	Credible reviews	Items	Users
TripAdvisor	800	800	20	-
Yelp	5169	37,500	273	24,769
Yelp*	5169	5169	151	7898

Table 3. Amazon dataset statistics for item ranking, with cumulative #items and varying #reviews.

Domain	#Users	#Reviews	#Items with reviews per-item						
			≤5	≤10	≤20	≤30	≤40	≤50	Total
Consumer electronics	94,664	121,234	14,797	16,963	18,350	18,829	19,053	19,187	19,518
Software	21,825	26,767	3,814	4,354	4,668	4,767	4,807	4,828	4,889
Sports	656	695	202	226	233	235	235	235	235

We use τ_m to find the optimal value of C^- by varying it in the interval $C^- \in \{0, 5, 10, 15, ...150\}$ using a *validation set* from Amazon as shown in Fig. 1. We observe that as C^- increases, τ_m also increases till a certain point as more and more non-credible reviews are filtered out, after which it stabilizes.

Ranking SVM – Our previous approach uses the model M_{Yelp} trained on Yelp, with the references sales ranking in Amazon being used only for evaluating the item rankings on the Kendall-Tau measure. To obtain a good item ranking based on credible reviews, a model M_{Amazon} that directly optimizes for Kendall-Tau using the reference ranking as training labels can be used. This allows the use of the entire feature space available in Amazon, including the explicit facet-label distribution and the full vocabulary. The feature space is constructed similarly to that of the Yelp dataset.

The goal of Ranking SVM [12] is to learn a ranking function which is concordant with a given ordering of items. The objective is to learn $\boldsymbol{w}$ such that $\boldsymbol{w} \cdot \boldsymbol{x_i} > \boldsymbol{w} \cdot \boldsymbol{x_j}$ for most data pairs $\{(\boldsymbol{x_i}, \boldsymbol{x_j}) : y_i > y_j \in R\}$. Although the problem is known to be NP-hard, it is approximated using SVM techniques with pairwise slack variables $\xi_{i,j}$. The optimization problem is equivalent to that of classifying SVM, but now operating on *pairwise difference vectors* $(\boldsymbol{x_i} - \boldsymbol{x_j})$ with corresponding labels $+1/-1$ indicating which one should be ranked ahead. We use the implementation of [12] (obtained from www.cs.cornell.edu/people/tj/svm_light/svm_rank.html) that maximizes the empirical Kendall-Tau by minimizing the number of discordant pairs.

Unlike the classification task, where labels are *per-review*, the ranking task requires labels *per-item*. Consider $\langle f_{i,j,k} \rangle$ to be the feature vector for the j^{th} review of an item i, with k indexing an element of the feature vector.

We aggregate these feature vectors element-wise over all the reviews on item i to obtain its feature vector $\langle \frac{\sum_j f_{i,j,k}}{\sum_j \mathbf{1}} \rangle$.

4 Experimental Setup

Parameter Initialization: The sentiment lexicon from [8] consisting of 2006 positive and 4783 negative polarity bearing words is used to initialize the review text based facet-label-word tensor Φ prior to inference. We consider the number of topics, $K = 20$ for Yelp, and $K = 50$ for Amazon with the review sentiment labels $L = \{+1, -1\}$ (corresponding to positive and negative rated reviews) initialized randomly. The symmetric Dirichlet priors are set to $\alpha = 50/K$, $\beta = 0.01$, and $\gamma = 0.1$.

Datasets and Ground-Truth: In this work, we consider the following datasets (refer to Tables 2 and 3) with available ground-truth information.

• The *TripAdvisor Dataset* [26,27] consists of 1600 reviews from TripAdvisor with positive (5 star) and negative (1 star) sentiment — comprising 20 credible and 20 non-credible reviews for *each* of 20 most popular Chicago hotels. The authors crawled the *credible* reviews from online review portals like TripAdvisor; whereas the *non-credible* ones were generated by users in Amazon Mechanical Turk. The dataset has only the review text and sentiment label (positive/negative ratings) with corresponding hotel names, with no other information on users or items.

• The *Yelp Dataset* consists of $37.5K$ recommended (i.e., *credible*) reviews, and $5K$ non-recommended (i.e., *non-credible*) reviews given by the Yelp filtering algorithm, on 273 restaurants in Chicago. For each review, we gather the following information: $\langle userId, itemId, timestamp, rating, review, metadata \rangle$. The metadata consists of some user activity information as outlined in Sect. 3.4.

The reviews marked as "not recommended" by the Yelp spam filter are considered to be the ground-truth for comparing the accuracy for credible review detection for our proposed model. The Yelp spam filter presumably relies on linguistic, behavioral, and social networking features [24].

• The *Amazon Dataset* used in [11] consists of around $149K$ reviews from nearly $117K$ users on $25K$ items from three domains, namely Consumer Electronics, Software, and Sports items. For each review, we gather the same information tuple as that from Yelp. However, the metadata in this dataset is not as rich as in Yelp, consisting only of helpfulness votes on the reviews.

Further, there exists no explicit ground-truth characterizing the reviews as credible or deceptive in Amazon. To this end, we re-rank the items using learning to rank, implicitly filtering out possible deceptive reviews (based on the feature vectors), and then compare the ranking to the *item sales rank* considered as the pseudo ground-truth.

Comparison Baselines: We use the following state-of-the-art baselines (given the full set of features that fit with their model) for comparison with our proposed model.

(1) Language Model Baselines: We consider the unigram and bigram language model baselines from [26,27] that have been shown to outperform other baselines using psycholinguistic features, part-of-speech tags, information gain, etc. We take the best baseline from their work which is a combination of unigrams and bigrams. Our proposed model (N-gram+Facet) enriches it by using length normalization, presence or absence of features, latent facets, etc. The recently proposed *doc-to-vec* model based on Neural Networks, overcomes the weakness of bag-of-words models by taking the context of words into account, and learns a dense vector representation for each document [13]. We train the doc-to-vec model in our dataset as a baseline model. In addition, we also consider readability (ARI) and review sentiment scores [9] under the hypothesis that writing styles would be random because of diverse customer background. ARI measures the reader's ability to comprehend a text and is measured as a function of the total number of characters, words, and sentences present, while review sentiment tries to capture the fraction of occurrences of positive/negative sentiment words to the total number of such words used.

(2) Activity and Rating Baselines: Given the tuple $\langle userId, itemId, rating, review, metadata\rangle$ from the Yelp dataset, we extract all possible activity and rating behavioral features of users as proposed in [10,11,14,17,22–24,32]. Specifically, we utilize the number of helpful feedbacks, review title length, review rating, use of brand names, percent of positive and negative sentiments, average rating, and rating deviation as features for classification. Further, based on the recent work of [29], we also use the user check-in and user elite status information as additional features for comparison.

Empirical Evaluations: Our experimental setup considers the following evaluations:

(1) Credible review classification: We study the performance of the various approaches in distinguishing a *credible* review from a *non-credible* one. Since this forms a binary classification task, we consider a balanced dataset containing equal proportion of data from each of the two classes. On the Yelp dataset, for each item we randomly sample an equal number of credible and non-credible reviews (to obtain Yelp*); while the TripAdvisor dataset is already balanced. Table 4 shows the 10-fold cross validation accuracy results for the different models on the two datasets. We observe that our proposed *consistency and behavioral features* exhibit around 15 % improvement in Yelp* for classification accuracy over the best performing baselines (refer to Table 4). Since the TripAdvisor dataset has *only* review text, the user/activity models could *not* be used there. The experiment could also not be performed on Amazon, as the ground-truth for credibility labels of reviews is absent.

(2) Item Ranking: In this task we examine the effect of non-credible reviews on the ranking of items in the community. This experiment is performed *only* on Amazon using the item *sales rank* as ground or reference ranking, as Yelp does not provide such item rankings. The sales rank provides an indication as to how

Table 4. Credible review classification accuracy with 10-fold cross validation. TripAdvisor dataset contains only review texts and no user/activity information.

Models	Features	TripAdvisor	Yelp*
Deep learning	Doc2Vec	69.56	64.84
	Doc2Vec + ARI + Sentiment	76.62	65.01
Activity & rating	Activity+Rating	-	74.68
	Activity+Rating+Elite+Check-in	-	79.43
Language	Unigram + Bigram	88.37	73.63
	Consistency	80.12	76.5
Behavioral	Activity Model$^-$	-	80.24
	Activity Model$^+$	-	86.35
Aggregated	N-gram + Consistency	**89.25**	79.72
	N-gram + Activity$^-$	-	82.84
	N-gram + Activity$^+$	-	88.44
	N-gram + Consistency + Activity$^-$	-	86.58
	N-gram + Consistency + Activity$^+$	-	**91.09**
	M_{Yelp}	-	89.87

well a product is selling on Amazon.com and highlights the item's rank in the corresponding category[2].

The baseline for the item ranking is based on the aggregated rating of all reviews on an item. The first model M_{Yelp} (C-SVM) trained on Yelp filters out the non-credible reviews, before aggregating review ratings on an item. The second model M_{Amazon} (SVM-Rank) is trained on Amazon using SVM-Rank with the reference ranking as training labels. 10-fold cross-validation results are reported on the two measures of Kendall-Tau (τ_b and τ_m) in Table 5 with respect to the reference ranking. τ_b and τ_m for SVM-Rank are the same since there are no ties. Our first model performs substantially better than the baseline, which, in turn, is outperformed by our second model.

In order to find the effectiveness of our approach in dealing with "long-tail" items, we perform an additional experiment with our best performing model i.e., M_{Amazon} (SVM-Rank). We use the model to find Kendall-Tau-M (τ_m) rank correlation (with the reference ranking) of items having less than (or equal to) 5, 10, 20, 30, 40, and 50 reviews in different domains in Amazon (results reported in Table 6 with 10-fold cross validation). We observe that our model performs substantially well even with items having as few as *five* reviews, with the performance progressively getting better with more reviews per-item.

[2] www.amazon.com/gp/help/customer/display.html?nodeId=525376.

Table 5. Kendall-Tau correlation of different models across domains.

Domain	Kendall-Tau-B (τ_b)		Kendall-Tau-M (τ_m)		Kendall-Tau ($\tau_b = \tau_m$)
	Baseline	M_{Yelp} (C-SVM)	Baseline	M_{Yelp} (C-SVM)	M_{Amazon} (SVM-Rank)
CE	0.011	0.109	0.082	0.135	0.329
Software	0.007	0.184	0.088	0.216	0.426
Sports	0.021	0.155	0.102	0.170	0.325

Table 6. Variation of Kendall-Tau-M (τ_m) correlation with #reviews with $M_{\textbf{Amazon}}$ **(SVM-Rank).**

Domain	τ_m with #reviews per-item						
	$\leq$5	$\leq$10	$\leq$20	$\leq$30	$\leq$40	$\leq$50	Overall
CE	0.218	0.257	0.290	0.304	0.312	0.317	0.329
Software	0.353	0.375	0.401	0.411	0.417	0.419	0.426
Sports	0.273	0.324	0.310	0.325	0.325	0.325	0.325

5 Discussions on Experimental Results

Language Model: The bigram language model performs very well (refer to Table 4) on the TripAdvisor dataset due to the setting of the task. Workers in Amazon Mechanical Turk were tasked with writing fake reviews with the guideline of knowing all the hotel amenities in its website before writing reviews. Therefore it is quite difficult for the facet model to find contradictions or mismatch in facet descriptions. Consequently, the facet model gives marginal improvement when combined with the language model.

On the other hand, the Yelp dataset is real-world, and therefore more noisy. The bigram language model and doc-to-vec hence do not perform as good as they do in the previous dataset; and neither does the facet model in isolation. However all the components put together give significant performance improvement over the ones in isolation (around 8 %).

Incorporation of writing style using ARI and sentiment measures improves performance of doc-to-vec in the TripAdvisor dataset, but not significantly in the real-world Yelp data.

Table 7 shows the top unigrams and bigrams contributing to the language feature space in the *joint model* for credibility classification — given by the feature weights of the C-SVM. We find that credible reviews contain a mix of function and content words, balanced opinions, with the highly contributing features being mostly unigrams. Whereas, non-credible reviews contain extreme opinions, less function words and more of sophisticated content words — consisting of a lot of signature bigrams — to catch the readers' attention.

Table 7. Top n-grams (by feature weights) for credibility classification.

Credible Reviews	Non-credible reviews
not, also, really, just, like, get, perfect, little, good, one, space, pretty, can, everything, come_back, still, us, right, definitely, enough, much, super, free, around, delicious, no, fresh, big, favorite, lot, selection, sure, friendly, way, dish, since, huge, etc., menu, large, easy, last, room, guests, find, location, time, probably, helpful, great, now, something, two, nice, small, better, sweet, though, loved, happy, love, anything, actually, home	dirty, mediocre, charged, customer_service, signature_lounge, view_city, nice_place, hotel_staff, good_service, never_go, overpriced, several_times, wait_staff, signature_room, outstanding, establishment, architecture_foundation, will_not, long, waste, food_great, glamour_closet, glamour, food_service, love_place, terrible, great_place, never, wonderful, atmosphere, signature, bill, will_never, good_food, management, great_food, money, worst, horrible, manager, service, rude

Behavioral Model: We find the activity based model to perform the best in isolation (refer to Table 4). Combined with language and consistency features, the joint model exhibits around 5 % improvement in performance. Additional metadata like the user elite and check-in status improves the performance of activity based baselines, which are not typically available for newcomers in the community. Our model using limited information (*N-gram+Consistency+Activity*$^-$) performs better than the activity baselines using fine-grained information about items (like brand description) and user history. Incorporating additional user features (*Activity*$^+$) further boosts its performance.

Consistency Features: In order to find the effectiveness of the facet based consistency features, we perform ablation tests (refer to Table 4). We remove the consistency model from the aggregated model, and see significant performance degradation of $3 - 4$ % for the Yelp* dataset. In the TripAdvisor dataset the performance reduction is less compared to Yelp due to reasons outlined before.

Table 8 shows a snapshot of the non-credible reviews, with corresponding (in)consistency features in Yelp and Amazon. We see that ratings of deceptive reviews do not corroborate with the textual description, irrelevant facets influencing the rating of the target item, contradicting other users, expressing extreme opinions without explanation, depicting temporal "burst" in ratings, etc. In principle, these features can also be used to detect other anomalous phenomena like group-spamming (one of the principal indicators of which is temporal burst), which is out of scope of this work.

Ranking Task: For the ranking task in Amazon (refer to Table 5), the first model M_{Yelp} — trained on Yelp and tested on Amazon using C-SVM — performs much better than the baseline exploiting various consistency features. The second model M_{Amazon} — trained on Amazon using SVM-Rank — outperforms the

Table 8. Snapshot of non-credible reviews (reproduced verbatim) with inconsistencies.

Inconsistency features	Yelp review & [rating]	Amazon review & [rating]
user review – rating *(promotion/demotion)*:	never been inside James. never checked in. never visited bar. yet, one of my favorite hotels in Chicago. James has dog friendly area. my dog loves it there. [5]	Excellant product-alarm zone, technical support is almost non-existent because of this i will look to another product. this is unacceptible. [4]
user review – facet description *(irrelevant)*:	you will learn that they are actually EVANGELICAL CHRISTIANS working to proselytize the coffee farmers they buy from. [2]	DO NOT BUY THIS. I used turbo tax since 2003, it never let me down until now. I can't file because Turbo Tax doesn't have software updates from the IRS "because of Hurricane Katrina". [1]
user review – item description *(deviation from community)*:	internet is charged in a 300 dollar hotel! [3]	The book Amazon offers is a joke! All it provides is the forward which is not written by Kalanithi. I don't have any sample of HIS writing to know if it appeals. [1]
extreme user rating:	GREAT!!!i give 5 stars!!!Keep it up. [5]	GREAT. This camera takes pictures. [1]
temporal bursts[3]**:**	Dan's apartment was beautiful and a great downtown location... (3/14/2012) [5] I highly recommend working with Dan and NSRA... (3/14/2012) [5] Dan is super friendly, demonstrating that he was confident... (3/14/2012) [5] my condo listing with no activity, Dan really stepped in... (4/18/2012) [5]	

[3]these reviews have also been flagged by the Yelp Spam Filter as not-recommended (i.e., non-credible)

former exploiting the power of the entire feature space and domain-specific proxy labels unavailable to the former.

"Long-Tail" Items: Table 6 shows the gradual degradation in performance of the second model M_{Amazon} (SVM-Rank) in dealing with items with lesser number of reviews. Nevertheless, we observe it to give a substantial Kendall-Tau correlation (τ_m) with the reference ranking, with as few as *five* reviews per-item, demonstrating the effectiveness of our model in dealing with "long-tail" items.

6 Conclusions

We present a novel consistency model using limited information for detecting non-credible reviews which is shown to outperform state-of-the-art baselines. Our approach overcomes the limitation of existing works that make use of fine-grained information which are not available for "long-tail" items or newcomers in the community. Most importantly prior methods are not designed to *explain* why the detected review should be non-credible. In contrast, we make use of different *consistency features* from *latent facet model* derived from user text and ratings that can explain the assessments by our method. We develop multiple models for domain transfer and adaptation, where our model performs very well in the ranking tasks involving "long-tail" items, with as few as *five* reviews per-item.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Chen, D.R., Wu, Q., Ying, Y., Zhou, D.X.: Support vector machine soft margin classifiers: error analysis. J. Mach. Learn. Res. **5**, 1143–1175 (2004)
3. Chen, Y.R., Chen, H.H.: Opinion spam detection in web forum:a real case study. In: WWW (2015)
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
5. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
6. Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Exploiting burstiness in reviews for review spammer detection. In: ICWSM (2013)
7. Feng, S., Banerjee, R., Choi, Y.: Syntactic stylometry for deception detection. In: ACL (2012)
8. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD (2004)
9. Hu, N., Bose, I., Koh, N.S., Liu, L.: Manipulation of online reviews: an analysis of ratings, readability, and sentiments. Decis. Support Syst. **52**(3), 674–684 (2012)
10. Jindal, N., Liu, B.: Analyzing and detecting review spam. In: ICDM, pp. 547–552 (2007)
11. Jindal, N., Liu, B.: Opinion spam and analysis. In: WSDM, pp. 219–230 (2008)
12. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD (2002)
13. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: ICML (2014)
14. Li, H., Chen, Z., Liu, B., Wei, X., Shao, J.: Spotting fake reviews via collective positive-unlabeled learning. In: ICDM, pp. 899–904 (2014)
15. Li, H., Chen, Z., Mukherjee, A., Liu, B., Shao, J.: Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In: ICWSM (2015)
16. Li, J., Ott, M., Cardie, C.: Identifying manipulated offerings on review portals. In: EMNLP (2013)
17. Lim, E., Nguyen, V., Jindal, N., Liu, B., Lauw, H.W.: Detecting product review spammers using rating behaviors. In: CIKM, pp. 939–948 (2010)
18. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: CIKM (2009)
19. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retrieval **3**(3), 225–331 (2009)
20. Luca, M., Zervas, G.: Fake it till you make it: Reputation, competition, and yelp review fraud. Technical report, Harvard Business School (2015)
21. Mihalcea, R., Strapparava, C.: The lie detector: explorations in the automatic recognition of deceptive language. In: ACL/IJCNLP (Short Papers), pp. 309–312 (2009)
22. Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., Ghosh, R.: Spotting opinion spammers using behavioral footprints. In: KDD, pp. 632–640 (2013)
23. Mukherjee, A., Liu, B., Glance, N.S.: Spotting fake reviewer groups in consumer reviews. In: WWW, pp. 191–200 (2012)
24. Mukherjee, A., Venkataraman, V., Liu, B., Glance, N.S.: What yelp fake review filter might be doing? In: ICWSM (2013)

25. Mukherjee, S., Weikum, G., Danescu-Niculescu-Mizil, C.: People on drugs: credibility of user statements in health communities. In: KDD, pp. 65–74 (2014)
26. Ott, M., Cardie, C., Hancock, J.T.: Negative deceptive opinion spam. In: NAACL (2013)
27. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: ACL-HLT, vol. 1. pp. 309–319 (2011)
28. Pennebaker, J., Francis, M., Booth, R.: Linguistic Inquiry and Word Count: A Computerized Text Analysis Program. Psychology Press, Mahwah (2001)
29. Rahman, M., Carbunar, B., Ballesteros, J., Chau, D.H.P.: To catch a fake: curbing deceptive yelp ratings and venues. Stat. Anal. Data Min. **8**(3), 147–161 (2015)
30. Strapparava, C., Valitutti, A.:WordNet-Affect: an affective extension of wordnet. In: LREC (2004)
31. Sun, H., Morales, A., Yan, X.: Synthetic review spamming and defense. In: KDD (2013)
32. Wang, G., Xie, S., Liu, B., Yu, P.S.: Review graph based online store review spammer detection. In: ICDM, pp. 1242–1247 (2011)
33. Yoo, K.H., Gretzel, U.: Comparison of deceptive and truthful travel reviews. In: ENTER (2009)