Identifying Manipulated Offerings on Review Portals

Jiwei Li
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
jiweil@cs.cmu.edu

Myle Ott **Claire Cardie**
Department of Computer Science
Cornell University
Ithaca, NY 14853, USA
myleott,cardie@cs.cornell.edu

Abstract

Recent work has developed supervised methods for detecting *deceptive opinion spam*—fake reviews written to sound authentic and deliberately mislead readers. And whereas past work has focused on identifying individual fake reviews, this paper aims to identify *offerings* (e.g., hotels) that contain fake reviews. We introduce a semi-supervised manifold ranking algorithm for this task, which relies on a small set of labeled individual reviews for training. Then, in the absence of gold standard labels (at an offering level), we introduce a novel evaluation procedure that ranks artificial instances of real offerings, where each artificial offering contains a known number of injected deceptive reviews. Experiments on a novel dataset of hotel reviews show that the proposed method outperforms state-of-art learning baselines.

1 Introduction

Consumers increasingly rely on user-generated online reviews when making purchase decisions (Cone, 2011; Ipsos, 2012). Unfortunately, the ease of posting content to the Web, potentially anonymously, combined with the public's trust and growing reliance on opinions and other information found online, create opportunities and incentives for unscrupulous businesses to post *deceptive opinion spam*—fraudulent or fictitious reviews that are deliberately written to sound authentic, in order to deceive the reader (Ott et al, 2011).

Unlike other kinds of spam, such as Web (Martinez-Romo and Araujo, 2009; Castillo et al, 2006) and e-mail spam (Chirita et al, 2005), recent work has found that deceptive opinion spam is neither easily ignored nor easily identified by human readers (Ott et al, 2011). Accordingly, there is growing interest in developing automatic, usually learning-based, methods to help users identify deceptive opinion spam (see Section 2). Even in fully-supervised settings, however, automatic methods are imperfect at identifying individual deceptive reviews, and erroneously labeling genuine reviews as deceptive may frustrate and alienate honest reviewers.

An alternative approach, not yet considered in previous work, is to instead identify those product or service offerings where fake reviews appear with high probability. For example, a hotel manager may post fake positive reviews to promote their own hotel, or fake negative reviews to demote a competitor's hotel. In both cases, rather than identifying these deceptive reviews individually, it may be preferable to identify the *manipulated offering* (i.e., the hotel) so that review portal operators, such as TripAdvisor or Yelp, can further investigate the situation without alienating users.[1]

Accordingly, this paper addresses the novel task of *identifying manipulated offerings*, which we frame as a ranking problem, where the goal is to rank offerings by the proportion of their reviews that are believed to be deceptive. We propose a novel three-layer graph model, based on manifold ranking (Zhou et al, 2003a; 2003b), to jointly model deceptive language at the offering-, review- and term-level. In particular, rather than treating reviews within the same offering as independent units, there is a reinforcing relationship between offerings and reviews.

[1] Manipulating online reviews may also have legal consequences. For example, the Federal Trade Commission (FTC) has updated their guidelines on the use of endorsements and testimonials in advertising to suggest that posting deceptive reviews may be unlawful in the United States (FTC, 2009).

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1933–1942,
Seattle, Washington, USA, 18-21 October 2013.

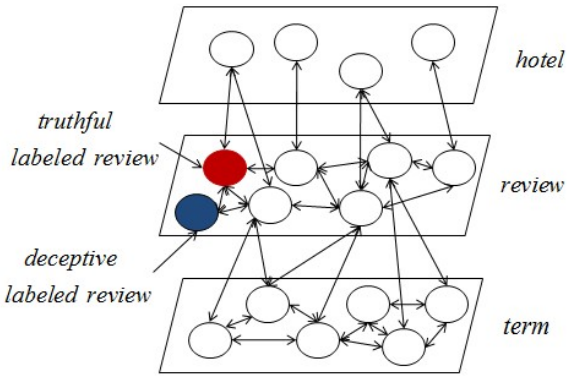

Figure 1: Mutual Reinforcement Graph Model for Hotel Ranking using the Manifold-Ranking Method

Our manifold ranking approach is semi-supervised in that it *requires no supervisory information at the offering level*; rather, it requires only a small amount of labeled data at a review level. Intuitively, and as depicted in Figure 1 for hotel offerings, we represent hotels, reviews and terms as nodes in a graph, where each hotel is connected to its reviews, and each review, in turn, is connected to the terms used within it. The influence of labeled data is propagated along the graph to unlabeled data, such that a hotel is considered more deceptive if it is heavily linked with other deceptive reviews, and a review, in turn, is more deceptive if it is generated by a deceptive hotel.

The success of our semi-supervised approach further depends on the ability to learn patterns of truthful and deceptive reviews that generalize across reviews of different offerings. This is challenging, because reviews often contain offering-specific vocabulary. For example, reviews of hotels in Los Angeles are more likely to include keywords such as "beach", "sea", "sunshine" or "LA", while reviews of Juneau hotels may contain "glacier", "Juneau", "bear" or "aurora borealis." A hotel review might also mention the hotel's restaurant or bar by name.

Unfortunately, it is unclear how important (or detrimental) offering-specific features are when deciding whether a review is fake. Accordingly, we *propose a dimensionality-reduction approach*, based on Latent Dirichlet Allocation (LDA) (Blei et al, 2003), *to obtain a vector representation of reviews for the ranking algorithm that generalizes across reviews of different offerings*. Specifically, we train an LDA-based topic model to view each review as a mixture of aspect-, city-, hotel- and review-specific topics (see Section 6). We then reduce the dimensionality of our data (i.e., labeled and unlabeled reviews) by replacing each review term vector with a vector that corresponds to its term distribution over just its aspect-specific topics, i.e., excluding city-, hotel- and review-specific topics. We find that, compared to models trained either on the full vocabulary, or trained on standard LDA document-topic vectors, this representation allows our models to generalize better across reviews of different offerings.

We evaluate our approach on the task of identifying (ranking) manipulated hotels. In particular, in the absence of gold standard offering-level labels, we introduce a *novel evaluation procedure for this task*, in which we rank numerous *versions* of each hotel, where each hotel version contains a different number of injected, known deceptive reviews. Thus, we expect hotel versions with larger proportions of deceptive reviews to be ranked higher than those with smaller proportions.

For labeled training data, we use the Ott et al. (2011) dataset of 800 *positive* (5-star) reviews of 20 Chicago hotels (400 deceptive and 400 truthful). For evaluation, *we construct a new* FOUR-CITIES *dataset*, containing 40 deceptive and 40 truthful reviews for each of eight hotels in four different cities (640 reviews total), following the procedure outlined in Ott et al. (2011). We find that our manifold ranking approach outperforms several state-of-the-art learning baselines on this task, including transductive Support Vector Regression. We additionally apply our approach to a large-scale collection of real-world reviews from TripAdvisor and explore the resulting ranking.

In the sections below, we discuss related work (Section 2) and describe the datasets used in this work (Section 3), the dimensionality-reduction approach for representing reviews (Section 4), and the semi-supervised manifold ranking approach (Section 5). We then evaluate the methods *quantitatively* (Sections 6 and 7) and *qualitatively* (Section 8).

2 Related Work

A number of recent approaches have focused on identifying individual fake reviews or users who post

fake reviews. For example, Jindal and Liu (2008) train machine learning classifiers to identify duplicate (or near duplicate) reviews. Yoo and Gretzel (2009) gathered 40 truthful and 42 deceptive hotel reviews and manually compare the psychologically relevant linguistic differences between them. Lim et al. (2010) propose an approach based on abnormal user behavior to predict spam users, without using any textual features. Ott et al. (2011) solicit deceptive reviews from workers on Amazon Mechanical Turk, and built a dataset containing 400 deceptive and 400 truthful reviews, which they use to train and evaluate supervised SVM classifiers. Ott et al. (2012) expand upon this work to estimate prevalences of deception in a review community. Mukherjee et al. (2012) study spam produced by groups of fake reviewers. Li et al. (2013) use topic models to detect differences between deceptive and truthful topic-word distributions. In contrast, in this work we aim to identify fake reviews at an offering level.[2]

LDA Topic Models. LDA topic models (Blei et al, 2003) have been employed for many NLP tasks in recent years. Here, we build on earlier work that uses topic models to (a) separate background information from information discussing the various "aspects" of products (e.g., Chemudugunta et al. (2007)) and (b) identify different levels of information (e.g., user-specific, location-specific, time-specific) (Ramage et al., 2009).

Manifold Ranking Algorithm. The manifold-ranking method (Zhou et al, 2003a; Zhou et al, 2003b) is a mutual reinforcement ranking approach initially proposed to rank data points along their underlying manifold structure. It has been widely used in many different ranking applications, such as summarization (Wan et al, 2007; Wan and Yang, 2007).

3 Dataset

In this paper, we train all of our models using the CHICAGO dataset of Ott et al (2011), which contains 20 deceptive and 20 truthful reviews from each of 20 Chicago hotels (800 reviews total). This dataset is unique in that it contains known (*gold standard*) deceptive reviews, solicited through Amazon Mechanical Turk, and is publicly-available.[3]

City	Hotels
Chicago	W Chicago, Palomar Chicago
New York	Hotel Pennsylvania, Waldorf Astoria
Los Angeles	Sheraton Gateway, The Westin Los Angeles Airport
Houston	Magnolia Hotel, Crowne Plaza Houston River Oaks

Table 1: Details of our FOUR-CITIES evaluation data.

Unfortunately, the CHICAGO dataset is limited, both in size (800 reviews) and scope, in that it only contains reviews of hotels in one city: Chicago. Accordingly, in order to perform a more realistic evaluation for our task, we construct a new dataset, FOUR-CITIES, that contains 40 deceptive and 40 truthful reviews from each of eight hotels in four different cities (640 reviews total).

We build the FOUR-CITIES dataset using the same procedure as Ott et al (2011), by creating and dividing 320 Mechanical Turk jobs, called *Human-Intelligence Tasks* (HITs), evenly across eight of the most popular hotels in our four chosen cities (see Table 1). Each HIT presents a worker with the name of a hotel and a link to the hotel's website. Workers are asked to imagine that they work for the marketing department of the hotel and that their boss has asked them to write a fake positive review, as if they were a customer, to be posted on a travel review website. Each worker is allowed to submit a single review, and is paid $1 for an acceptable submission.

Finally, we augment our deceptive FOUR-CITIES reviews with a matching set of truthful reviews from TripAdvisor by randomly sampling 40 positive (5-star) reviews for each of the eight chosen hotels. While we cannot know for sure that the sampled reviews are truthful, previous work has suggested that rates of deception among popular hotels is likely to be low (Jindal and Liu, 2008; Lim et al, 2010).

4 Topic Models for Dimensionality Reduction

As mentioned in the introduction, we want to learn patterns of truthful and deceptive reviews that apply

[2] Approaches for identifying individual fake reviews may be applied to our task, for example, by averaging the review-level predictions for an offering. This averaging approach is one of our baselines in Section 7.

[3] We use the dataset available at: `http://www.cs.cornell.edu/~myleott/op_spam`.

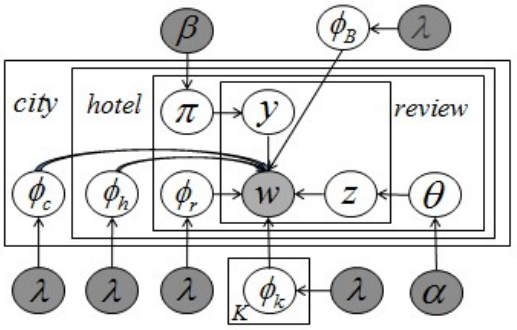

Figure 2: Graphical illustration of the RLDA topic model.

across hotels in different locations. This is challenging, however, because hotel reviews often contain specific information about the hotel or city, and it is unclear whether these features will generalize to reviews of other hotels.

We therefore investigate an LDA-based dimensionality-reduction approach (RLDA) to derive effective vector representations of reviews. Specifically, we model each document as a bag of words, generated from a mixture of: (a) "aspect" topics (that discuss various dimensions of the offering); (b) city-specific topics; (c) hotel-specific topics; (d) review-specific topics;[4] and (e) a background topic. We use this model to reduce the dimensionality of the review representation in our training and test sets, by replacing each review's term vector with a vector corresponding to the distribution over only the aspect-based topics, i.e., we exclude city, hotel and review-specific topics, as well as the background topic.

Below we present specific details of our model (Sections 4.1 and 4.2). The effectiveness of our dimensionality-reduction approach will be directly evaluated in Section 6, by comparing the performance of various classifiers trained either on the full vocabulary, or on our reduced feature representation.

4.1 RLDA Model Details

The plate diagram and generative story for our model are given in Figures 2 and 3, respectively. Our model has a similar general structure to standard LDA, but with additional machinery to handle different levels of information. In particular, in order to model K aspects in a collection of R reviews,

[4]These will be terms used in just a small number of reviews.

- Draw $\phi_B \sim Dir(\lambda)$
- For each aspect $z = 1, 2, ..., K$: draw $\phi^z \sim Dir(\lambda)$
- For each city $c = 1, 2, ..., C$: draw $\phi^c \sim Dir(\lambda)$
- For each hotel $h = 1, 2, ..., H$: draw $\phi^h \sim Dir(\lambda)$
- For each review r:
  - Draw $\pi^r \sim Dir(\beta)$
  - Draw $\phi^r \sim Dir(\lambda)$
  - Draw $\theta^r \sim Dir(\alpha)$
  - For each word w in d:
    - Draw $y_w \sim Multi(\pi_r)$
    - If $y_w = 0$:
      - Draw $z_w \sim Multi(\theta)$
      - Draw $w \sim Multi(\phi^{z_w})$
    - If $y_w = 1$: draw $w \sim Multi(\phi_B)$
    - If $y_w = 2$: draw $w \sim Multi(\phi_d)$
    - If $y_w = 3$: draw $w \sim Multi(\phi_h)$
    - If $y_w = 4$: draw $w \sim Multi(\phi_c)$

Figure 3: Generative story for the RLDA topic model.

of H hotels, in C cities, we first draw multinomial word distributions corresponding to: the background topic, ϕ_B; aspect topics, ϕ_k for $k \in [1, K]$; review-specific topics, ϕ_r for $r \in [1, R]$; hotel-specific topics, ϕ_h for $h \in [1, H]$; and city-specific topics, ϕ_c for $c \in [1, C]$. Then, for each word w in review R, we sample a switch variable, $y \in [0, 4]$, indicating whether w comes from one of the aspect topics ($y = 0$), or the background topic ($y = 1$), review-specific topic ($y = 2$), hotel-specific topic ($y = 3$) or city-specific topic ($y = 4$). If the word comes from one of the aspect topics, then we further sample the specific aspect topic, $z_w \in [1, K]$. Finally, we generate the word, w, from the corresponding ϕ.

4.2 Inference for RLDA

Given the review collection, our goal is to find the most likely assignment y_w (and z_w if $y_w = 0$) for each word, w, in each review. We perform inference using Gibbs sampling. It is relatively straightforward to derive Gibbs sampling equations that allow joint sampling of the z_w and y_w latent variables for each word token w:

$$P(y_w = 0, Z_w = k) = \frac{N^a_{r,-w} + \beta}{N_{r,-w} + 5\beta} \times \frac{C^k_{r,-w} + \alpha}{\sum_k C^k_{r,-w} + K\alpha} \times \frac{E^w_k + \lambda}{\sum_w E^w_k + V\lambda},$$

$$P(y_w = m, m = 1, 2, 3, 4) = \frac{N^m_{r,-w} + \beta}{N_{r,-w} + 5\beta} \times \frac{E^w_m + \lambda}{\sum_w E^w_m + V\lambda},$$

Note that the subscript $-w$ indicates that the count for word token w is excluded. Also, N_r

denotes the number of words in review r and $N^a_{r,-w}, N^1_{r,-w}, N^2_{r,-w}, N^3_{r,-w}, N^4_{r,-w}$ are the number of words in review r assigned to the aspect, background, review-specific, hotel-specific and city-specific topics, respectively, excluding the current word. $C^k_{r,-w}$ denotes the number of words in review r assigned to aspect topic k. $E^w_k, E^w_1, E^w_2, E^w_3, E^w_4$ denote the number of times that the word w is assigned to aspect k, the background topic, review-specific topic r, hotel-specific topic h, and city-specific topic c, respectively. We set hyperparameter α to 1, β to 0.5, λ to 0.01. We run 200 iterations of Gibbs sampling until the topic distribution stabilizes. After each iteration in Gibbs sampling, we obtain:

$$\pi^i_r = \frac{N^i_r + \beta}{\sum_i N^i_r + 5\beta} \quad \theta^k_r = \frac{C^k_r + \alpha}{\sum_k C^k_r + K\alpha}$$
$$\phi^w_z = \frac{E^w_z + \lambda}{\sum_w E^w_z + V\lambda} \quad \phi^w_m = \frac{E^w_m + \lambda}{\sum_w E^w_m + V\lambda} \tag{1}$$

Finally, at the end of Gibbs sampling, we filter out background, document-specific, hotel-specific and city-specific information, by replacing each document's term vector with a $1 \times K$ aspect-topic vector, $\vec{G_r} = \langle \theta^1_r, \theta^2_r, \cdots, \theta^K_r \rangle$.

5 Manifold Ranking for Hotels

In this section, we describe our ranking algorithm — based on manifold ranking (Zhou et al, 2003a; Zhou et al, 2003b) — that tries to jointly model deceptive language at the hotel-, review- and term-level.

5.1 Graph Construction

We use a three-layer (hotel layer, review layer and term layer) mutual reinforcement model (see Figure 1). Formally, we represent our three-layer graph as $G = \langle V_H, V_R, V_T, E_{HR}, E_{RR}, E_{RT}, E_{TT} \rangle$, where $V_H = \{H_i\}_{i=1}^{i=N_H}$, $V_R = \{R\}_{i=1}^{i=H_R}$ and $V_T = \{T_i\}_{i=1}^{i=V}$ correspond to the set of hotels, reviews and terms respectively. E_{HR}, E_{RR} and E_{RT} respectively denote the edges between hotels and reviews, reviews and reviews and reviews and terms. Each edge is associated with a weight that denotes the similarity between two nodes.

Let $sim(H_i, R_j)$, where $H_i \in V_H$ and $R_j \in V_R$, denote the edge weight between hotel H_i and review R_j, calculated as follows:

$$sim(H_i, R_j) = \begin{cases} 1 & if \quad R_i \in H_j \\ 0 & if \quad R_i \notin H_j \end{cases} \tag{2}$$

Then we get row normalized matrices $D_{HR} \in \mathbb{R}^{N_H \times N_R}$ and $D_{RH} \in \mathbb{R}^{N_R \times N_H}$ as follows:

$$D_{HR}(i,j) = \frac{sim(H_i, R_j)}{\sum_{i'} sim(H_{i'}, R_j)}$$
$$D_{RH}(i,j) = \frac{sim(H_i, R_j)}{\sum_{j'} sim(H_i, R_{j'})} \tag{3}$$

As described in Section 4.2, each review is represented with a $1 \times K$ aspect vector G_r after filtering undesired information. The edge weight between two reviews is then the cosine similarity, $sim(R_i, R_j)$, between two reviews and can be calculated as follows:

$$sim(R_i, R_j) = \frac{\sum_{t=1}^{t=K} G^t_i \cdot G^t_j}{\sqrt{\sum_{t=1}^{t=K} G^{t2}_i} \cdot \sqrt{\sum_{t=1}^{t=K} G^{t2}_j}} \tag{4}$$

Since the normalization process will make the review-to-review relation matrix asymmetric, we adopt the following strategy: let P denote the similarity matrix between reviews, where $P(i,j) = sim(R_i, R_j)$ and M denotes the diagonal matrix with (i,i)-element equal to the sum of the i^{th} row of SIM. The normalized matrix between reviews $D_{RR} \in \mathbb{R}^{N_R \times N_R}$ is calculated as follows:

$$D_{RR} = M^{-\frac{1}{2}} \cdot P \cdot M^{-\frac{1}{2}} \tag{5}$$

$sim(R_i, w_j)$ denotes the similarity between review R_i and term w_j and is the conditional probability of word w_j given review R_i. If $w_j \in R_j$, $sim(R_i, w_j)$ is calculated according to Eq. (6) by integrating out latent parameters θ and π. Else if $w_j \notin R_j$, $sim(R_i, w_j) = 0$.

$$\begin{aligned} sim(R_i, w_j) &= \sum_{k=1}^{k=K} p(z=k|r_i) \times p(w_j|z=k) \\ &\quad + \sum_{t \in \{B,h,c,d\}} p(w_j|y_i=t) p(y_i=t|r_i) \\ &= \pi^{(a)}_d \sum_{k=1}^{k=K} \theta^z_d \cdot \phi^{(w_j)}_z + \sum_{t \in \{B,h,c,d\}} \pi^{(t)}_d \phi^{(w_j)}_t \end{aligned} \tag{6}$$

Similar to Eq. (3), we further get the normalized matrix $D_{RT} \in \mathbb{R}^{H_R \times V}$ and $D_{TR} \in \mathbb{R}^{V \times H_R}$.

Similarity between terms $sim(w_i, w_j)$ is given by the WordNet path-similarity,[5] normalized to create the matrix D_{VV}.

[5]Path-similarity is based on the shortest path that connects the senses in the "is-a" (hypernym/hyponym) taxonomy. See `http://nltk.googlecode.com/svn/trunk/doc/howto/wordnet.html`.

Input: The hotel set V_D, review set V_R, term set V_T, normalized transition probability matrix $D_{HR}, D_{RR}, D_{RH}, D_{RT}, D_{TT}, D_{TR}$.
Output: the ranking vectors S_R, S_H, S_T.
Begin:

1. Initialization: set the score labeled reviews to $+1$ or -1 and other unlabeled reviews 0: $S_R^0 = [+1, ..., +1, -1, ..., -1, 0, ..., 0]$. Set S_H^0 and S_T^0 to 0. Normalize the score vector.
2. update S_R^k, S_H^k and S_T^k according to Eq. (7).
3. normalize S_R^k, S_H^k and S_T^k.
4. fix the score of labeled reviews to $+1$ and -1. Go to step (2) until convergence.

Figure 4: Semi-Supervised Reinforcement Ranking.

5.2 Reinforcement Ranking Based on the Manifold Method

Based on the set of labeled reviews, nodes for truthful reviews (positive) are initialized with a high score (1) and nodes for deceptive reviews, a low score (-1). Given the weighted graph, our task is to assign a score to the each hotel, each term, and the remaining unlabeled reviews. Let S_H, S_R and S_T denote the ranking scores of hotels, reviews and terms, which are updated during each iteration as follows until convergence[6]:

$$\begin{cases} S_H^{k+1} = D_{HR} \cdot S_R^k \\ S_R^{k+1} = \epsilon_1 D_{RR} \cdot S_R^k + \epsilon_2 D_{RH} \cdot S_H^k + \epsilon_3 D_{RT} \cdot S_t^k \\ S_T^{k+1} = \epsilon_4 D_{TT} \cdot S_T^k + \epsilon_5 D_{TR} \cdot S_R^k \end{cases} \tag{7}$$

where $\epsilon_1 + \epsilon_2 + \epsilon_3 = 1$ and $\epsilon_4 + \epsilon_5 = 1$. (The score of labeled reviews will be fixed to $+1$ or -1.)

6 Learning Generalizable Classifiers

In Section 4, we introduced RLDA to filter out review-, hotel- and city-specific information from our vector-based review representation. Here, we will directly evaluate the effectiveness of RLDA by comparing the performance of binary deceptive vs. truthful classifiers trained on three feature sets: (a) the full vocabulary, encoded as unigrams and bigrams (N-GRAMS); (b) a reduced-dimensionality feature space, based on standard LDA (Blei et al, 2003); and (c) a reduced-dimensionality feature space, based on our proposed revised LDA approach (RLDA).

We compare two kinds of classifiers, which are trained on only the labeled CHICAGO dataset and tested on the FOUR-CITIES dataset. First, we use SVM^{light} (Joachims, 1999) to train linear SVM classifiers, which have been shown to perform well in related work (Ott et al, 2011). Second, we train a two-layer manifold classifier, which is a simplified version of the model presented in Section 5. In this model, the graph consists of only review and term layers, and the score of a labeled review is fixed to 1 or -1 in each iteration. After convergence, reviews with scores greater than 0 are classified as truthful, and less than 0 as deceptive.

Results and Discussion The results are shown in Table 2 and show the average accuracy and precision/recall w.r.t. the *truthful* (positive) class. We find that SVM and MANIFOLD are comparable in all six conditions, and not surprisingly, perform best when evaluated on reviews from the two Chicago hotels in our FOUR-CITIES data. However, the N-GRAM and LDA feature sets perform much worse than RDLA when evaluation is performed on reviews from the other three (non-Chicago) cities. This confirms that classifiers trained on n-gram features overfit to the training data (CHICAGO) and do not generalize well to reviews from other cities. In addition, the standard LDA-based method for dimensionality reduction is not sufficient for our specific task.

7 Identifying Manipulated Hotels

In this section, we evaluate the performance of our manifold ranking approach (see Section 5) on the task of identifying *manipulated hotels*.

Baselines. We consider several baseline ranking approaches to compare to our manifold ranking approach. Like the manifold ranking approach, the baselines also employ both the CHICAGO dataset (labeled) and FOUR-CITIES dataset (**without** labels).[7] For fair comparison, we use identical processing techniques for each approach. Topic number is set

[6] Convergence is achieved if the difference between ranking scores in two consecutive iterations is less than 0.00001.

[7] While we have not investigated the effects of unlabeled data in detail, providing additional unlabeled data (beyond the test set) boosts the manifold ranking performances reported below by 1-2%.

city	feature set	SVM			Manifold		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall
Chicago	N-GRAMS	0.831	0.844	0.818	0.835	0.844	0.825
	LDA	0.833	0.846	0.819	0.817	0.832	0.802
	RLDA	0.830	0.838	0.822	0.841	0.819	0.863
Non-Chicago	N-GRAMS	0.728	0.744	0.714	0.733	0.738	0.727
	LDA	0.714	0.696	0.732	0.728	0.715	0.741
	RLDA	0.791	0.799	0.780	0.801	0.787	0.815

Table 2: Binary classification results showing that n-gram features overfit to the CHICAGO training data. Results correspond to evaluation on reviews for the two Chicago hotels from FOUR-CITIES and non-Chicago FOUR-CITIES reviews (six hotels).

to five for all topic-model-based approaches. Each baseline makes review-level predictions and then ranks each hotel by the average of those predictions.

- **Review-SVR:** Uses linear Tranductive Support Vector Regression with unigram and bigram features, similar to Ott et al. (2011).
- **Review-SVR+LDA (R):** Similar to REVIEW-SVR but uses our revised LDA (RLDA) topic model for dimensionality reduction **(R)**.
- **Two-Layer Manifold (S):** A simplified version of our model where the hotel-level is removed from the graph. Dimensionality reduction is performed using standard LDA **(S)**.
- **Two-Layer Manifold (R):** Similar to TWO-LAYER MANIFOLD (S) but uses the revised LDA (RLDA) model for dimensionality reduction.
- **Three-layer Manifold (tf-idf):** Our three-layer manifold ranking model, except with each review represented as a TF-IDF term vector. Review similarity is calculated based on the cosine similarity between these vectors.

Evaluation Method. To evaluate ranking performance in the absence of a gold standard set of manipulated hotels, we rearrange the FOUR-CITIES test set of 40 truthful and 40 deceptive reviews for each of eight hotels: we create 41 *versions* of each hotel, where each hotel version contains a different number of injected deceptive reviews, ranging from 0 to 40. For example, the first version of a hotel will have 40 truthful and 0 deceptive reviews, the second version 39 truthful and 1 deceptive, and the 41st version 0 truthful and 40 deceptive. In total, we generate $41 \times 8 = 328$ versions of hotel reviews. We expect versions with larger proportions of deceptive reviews to receive lower scores by the ranking models (i.e., they are ranked higher/more deceptive).

Metrics. To qualitatively evaluate the ranking results, we use the Normalized Discounted Cumulative Gain (NDCG), which is commonly used to evaluate retrieval algorithms with respect to an ideal relevance-based ranking. In particular, NDCG rewards rankings with the most relevant results at the top positions (Liu, 2009), which is also our objective, namely, to rank versions that have higher proportions of deceptive reviews nearer to the top.

Let $R(m)$ denote the relevance score of m^{th} ranked hotel version. Then, NDCG_N is defined as:

$$NDCG_N = \frac{1}{IDCG_N} \sum_{m=1}^{m=N} \frac{2^{R(m)} - 1}{log_2(1+m)} \quad (8)$$

where $IDCG_N$ refers to discounted cumulative gain (DCG) of the ideal ranking of the top N results. We define the ideal ranking according to the proportion of deceptive reviews in different versions, and report NDCG scores for the Nth ranked hotel versions ($N = 8$ to 321), at intervals of 8 (to account for ties among the eight hotels).

Results and Discussion. NDCG results are shown in Figure 5. We observe that our approach (using 2, 5 or 10 topics) generally outperforms the other approaches. In particular, approaches that use our RLDA text representation (OUR APPROACH, TWO-LAYER MANIFOLD (R), and REVIEW-SVR+LDA (R)), which tries to remove city- and hotel-specific information, perform better than those that use the full vocabulary (REVIEW-SVR, TWO-LAYER MANIFOLD (S), and THREE-LAYER MANIFOLD (TF-IDF)). This further confirms that our RLDA dimensionality reduction technique allows models,

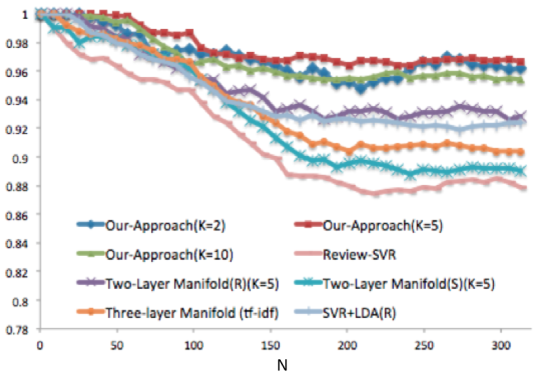

Figure 5: NDCG_N results for different approaches. K indicates the number of topics.

trained on limited data, to generalize to reviews of different hotels and in different locations. We also find that approaches that model a reinforcing relationship between hotels and their reviews are better than approaches that model reviews as independent units, e.g., TWO-LAYER MANIFOLD (R) vs. REVIEW-SVR+LDA and TWO-LAYER MANIFOLD (S) vs. REVIEW-SVR. This confirms our intuition that a hotel is more deceptive if it is connected with many deceptive reviews, and, in turn, a review is more deceptive if from a deceptive hotel.

8 Qualitative Evaluation

We now present qualitative evaluations for the RLDA topic model and the manifold ranking model.

Topic Quality. Table 3 gives the top words for four aspect topics and four city-specific topics in the RLDA topic model; Table 4 gives the highest and lowest ranking term weights in our three-layer manifold model. By comparing the first row of topics in Table 3, corresponding to aspect topics, to the top words in Table 4, we observe that the learned topics relate to truthful and deceptive classes. For example, Topics 1 and 4 share many terms with the top truthful terms in the manifold model, e.g., spatial terms, such as `location`, `floor` and `block`, and punctuation, such as `(`, `)`, and `$`. Similarly, Topics 2 and 7 share many terms with the top deceptive terms in the manifold model, e.g., `hotel`, `husband`, `wife`, `amazing`, `experience` and `recommend`. This makes sense, since topic models have been shown to produce discriminative topics on this data in previous work (Li et al., 2013).

Topic1	**Topic2**	**Topic4**	**Topic7**
location	hotel	(	hotel
$	stay	room	service
walk	staff	)	husband
night	restaurant	park	amazing
block	friendly	bed	will
floor	room	night	weekend
quiet	recommend	shower	friendly
nice	love	view	travel
lobby	excellent	minute	experience
breakfast	wife	pillow	friend
NYC	**Chicago**	**LA**	**Houston**
York	Chicago	los	Houston
ny	Michigan	Angeles	downtown
time	mile	la	Texas
square	tower	lax	cab
nyc	Illinois	shuttling	Westside
street	avenue	hollywood	center
empire	Rogers	plane	Northwest
Chinatown	river	morning	st
station	Burnham	California	museum
Wall	Goodman	downtown	mission

Table 3: Top words in topics extracted from RLDA topic model (see Section 4). The top row presents topic words from four aspect topics ($K = 10$) and the bottom row presents top words from four city-specific topics.

Deceptive		**Truthful**	
term	**score**	**term**	**score**
my	-1.063	$	0.964
visit	-0.944	location	0.922
we	-0.882	(	0.884
hotel	-0.863	)	0.884
husband	-0.828	bathroom	0.842
family	-0.824	floor	0.810
amazing	-0.782	breakfast	0.784
experience	-0.740	bar	0.762
recommend	-0.732	block	0.747
wife	-0.680	small	0.721
relax	-0.678	but	0.720
vacation	-0.651	walk	0.707
will	-0.651	lobby	0.707
friendly	-0.646	quiet	0.684

Table 4: Term scores from our ranking algorithm.

With respect to the second row in Table 4, containing top words from city-specific topics, we observe that each topic does contain primarily city-specific information. This helps to explain why removing terms associated with these topics resulted in a better vector representation for reviews.

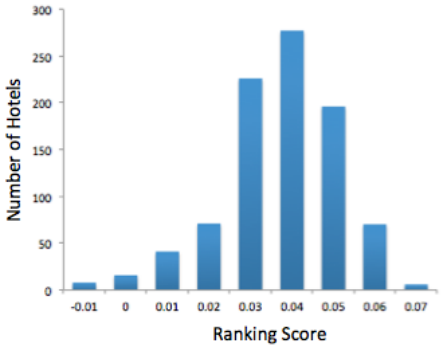

Figure 6: Hotel Ranking Distribution on TripAdvisor

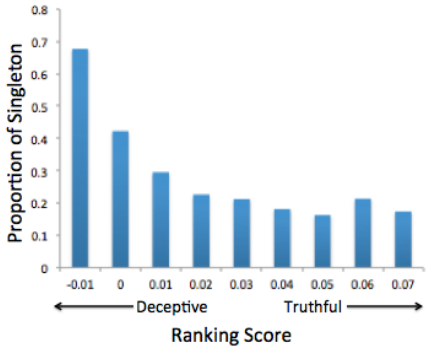

Figure 7: Proportion of Singletons vs. Hotel Ranking.

Real-world Evaluation. Finally, we apply our ranking model to a large-scale collection of real-world reviews from TripAdvisor. We crawl 878,561 reviews from 3,945 hotels in 25 US cities from TripAdvisor excluding all non-5-star reviews and removing hotels with fewer than 100 reviews. In the end, we collect 244,810 reviews from 838 hotels.

We apply our manifold ranking model and rank all 838 hotels. First, we present a histogram of the resulting manifold ranking scores in Figure 6. We observe that the distribution reaches a peak around 0.04, which in our quantitative evaluation (Section 7) corresponded to a hotel with 34 truthful and 6 deceptive reviews. These results suggest that the majority of reviews in TripAdvisor are truthful, in line with previous findings by Ott et al. (2011).

Next, we note that previous work has hypothesized that deceptive reviews are more likely to be posted by first-time review writers, or *singleton* reviewers (Ott et al, 2011; Wu et al, 2011). Accordingly, if this hypothesis were valid, then manipulated hotels would have an above-average proportion of singleton reviews. Figure 7 shows a histogram of the average proportion of singleton reviews, as a function of the ranking scores produced by our model. Noting that lower scores correspond to a higher predicted proportion of deceptive reviews, we observe that hotels that are ranked as being more deceptive by our model have much higher proportions of singleton reviews, on average, compared to hotels ranked as less deceptive.

9 Conclusion

We study the problem of identifying manipulated offerings on review portals and propose a novel three-layer graph model, based on manifold ranking for ranking offerings based on the proportion of reviews expected to be instances of deceptive opinion spam. Experimental results illustrate the effectiveness of our model over several learning-based baselines.

Acknowledgments

This work was supported in part by National Science Foundation Grant BCS-0904822, a DARPA Deft grant, as well as a gift from Google. We also thank the EMNLP reviewers for their helpful comments and advice.

References

David Blei, Ng Andrew and Michael Jordan. Latent Dirichlet allocation. 2003. In *Journal of Machine Learning Research*.

Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini and Sebastiano Vigna. A reference collection for web spam. In *ACM Sigir Forum*. 2006.

Paul-Alexandru Chirita, Jrg Diederich and Wolfgang Nejdl. MailRank: using ranking for spam detection. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. 2005.

Cone. 2011 Online Influence Trend Tracker. http://www.coneinc.com/negative-reviews-online-reverse-purchase-decisions. August.

Yajuan Duan, Zhumin Chen, Furu Wei, Ming Zhou and Heung-Yeung Shum. Twitter Topic Summarization by Ranking Tweets Using Social Influence and Content Quality. In *Proceedings of 24th International Conference on Computational Linguistics* 2012.

Federal Trade Commission. Guides Concerning Use of Endorsements and Testimonials in Advertising. In *FTC 16 CFR Part 255*. 2009.

Socialogue: Five Stars? Thumbs Up? A+ or Just Average? URL:http://www.ipsos-na.com/news-polls/pressrelease.aspx?id=5929g

Nitin Jindal, and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 2008.

Nitin Jindal, Bing Liu and Ee-Peng Lim. Finding Unusual Review Patterns Using Unexpected Rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*.2010.

Thorsten Joachims. Making large-scale support vector machine learning practical. In *Advances in kernel methods.*1999.

Fangtao Li, Minlie Huang, Yi Yang and Xiaoyan Zhu. Learning to identify review Spam. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*. 2011.

Jiwei Li, Claire Cardie and Sujian Li. TopicSpam: a Topic-Model-Based Approach for Spam Detection. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguis- tics.* 2013.

Peng Li, Jing Jiang and Yinglin Wang. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* 2010.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting Product Review Spammers Using Rating Behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 2010.

Tieyan Liu. Learning to Rank for Information Retrieval. In *Foundations and Trends in Information Retrieval* 2009.

Arjun Mukherjee, Bing Liu and Natalie Glance. Spotting Fake Reviewer Groups in Consumer Reviews . In *Proceedings of the 21st international conference on World Wide Web.* 2012.

Juan Martinez-Romo and Lourdes Araujo. Web spam identification through language model analysis. In *Proceedings of the 5th international workshop on adversarial information retrieval on the web.* 2009.

Myle Ott, Claire Cardie and Jeffrey Hancock. Estimating the Prevalence of Deception in Online Review Communities. In *Proceedings of the 21st international conference on World Wide Web*. 2012.

Myle Ott, Yejin Choi, Claire Cardie and Jeffrey Hancock. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.* 2011.

Daniel Ramage, David Hall, Ramesh Nallapati and Christopher Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing.* 2009.

Michal Rosen-zvi, Thomas Griffith, Mark Steyvers and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence.*2004.

Xiaojun Wan and Jianwu Yang. Multi-Document Summarization Using Cluster-Based Link Analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.* 2008.

Xiaojun Wan, Jianwu Yang and Jianguo Xiao Manifold-Ranking Based Topic-Focused Multi-Document Summarization. In *Proceedings of International Joint Conferences on Artificial Intelligence*,2007.

Guan Wang, Sihong Xie, Bing Liu and Philip Yu. Review Graph based Online Store Review Spammer Detection. In *Proceedings of International Conference of Data Mining.* 2011.

Guangyu Wu, Derek Greene and , Padraig Cunningham. Merging multiple criteria to identify suspicious reviews. In *Proceedings of the fourth ACM conference on Recommender systems.* 2011.

Kyung-Hyan Yoo and Ulrike Gretzel. Comparison of Deceptive and Truthful Travel Reviews. In *Information and Communication Technologies in Tourism.* 2009.

Dengyong Zhou, Olivier Bousquet, Thomas Navin and Jason Weston. Learning with local and global consistency. In *Proceedings of Advances in neural information processing systems.*2003.

Dengyong Zhou, Jason Weston, Arthur Gretton and Olivier Bousquet. Ranking on data manifolds. In *Proceedings of Advances in neural information processing systems.*2003.