IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL:26 NO:7 YEAR 2014

Computing Multi-Dimensional Trust by Mining E-Commerce Feedback Comments

Xiuzhen Zhang, Lishan Cui, and Yan Wang, *Senior Member, IEEE*

Abstract—Reputation-based trust models are widely used in e-commerce applications, and feedback ratings are aggregated to compute sellers' reputation trust scores. The "all good reputation" problem however is prevalent in current reputation systems – reputation scores are universally high for sellers and it is difficult for potential buyers to select trustworthy sellers. In this paper, based on the observation that buyers often express opinions openly in free text feedback comments, we propose CommTrust for trust evaluation by mining feedback comments. Our main contributions include: (1) We propose a multidimensional trust model for computing reputation scores from user feedback comments; (2) We propose an algorithm for mining feedback comments for dimension ratings and weights, combining techniques of natural language processing, opinion mining and topic modelling. Extensive experiments on eBay and Amazon data demonstrate that CommTrust can effectively address the "all good reputation" issue and rank sellers effectively. To the best of our knowledge, our research is the first piece of work on trust evaluation by mining feedback comments.

Index Terms—K.4.4 Electronic commerce; H.2.8.I Text mining.

1 Introduction

Accurate trust evaluation is crucial for the success of e-commerce systems. Reputation reporting systems [1] have been implemented in e-commerce systems such as eBay and Amazon (for third-party sellers), where overall reputation scores for sellers are computed by aggregating feedback ratings. For example on eBay, the reputation score for a seller is the *positive percentage score*, as the percentage of positive ratings out of the total number of positive ratings and negative ratings in the past 12 months. [1]

A well-reported issue with the eBay reputation management system is the "all good reputation" problem [1], [2] where feedback ratings are over 99% positive on average [1]. Such strong positive bias can hardly guide buyers to select sellers to transact with. At eBay detailed seller ratings for sellers (DSRs) on four aspects of transactions, namely *item as described*, *communication*, *postage time*, and *postage and handling charges*, are also reported. DSRs are aggregated rating scores on a 1- to 5-star scale. Still the strong positive bias is present – aspect ratings are mostly 4.8 or 4.9 stars. One possible reason for the lack of negative ratings at e-commerce web sites is that users who leave negative feedback ratings can attract retaliatory negative ratings and thus damage their own reputation [1].

Although buyers leave positive feedback ratings, they express some disappointment and negativeness in free text feedback comments [3], often towards specific aspects of transactions. For example, a comment like "*The products were as I expected.*" expresses positive opinion towards the *product* aspect, whereas the comment "*Delivery was a little slow but otherwise, great service. Recommend highly.*" expresses negative opinion towards the *delivery* aspect but a positive opinion to the *transaction* in general. By analysing the wealth of information in feedback comments we can uncover buyers' detailed embedded opinions towards different aspects of transactions, and compute comprehensive reputation profiles for sellers.

We propose *Comment-based Multi-dimensional trust (CommTrust)*, a fine-grained multi-dimensional trust evaluation model by mining e-commerce feedback comments. With CommTrust, comprehensive trust profiles are computed for sellers, including dimension reputation scores and weights, as well as overall trust scores by aggregating dimension reputation scores. To the best of our knowledge, CommTrust is the first piece of work that computes fine-grained multidimension trust profiles automatically by mining feedback comments. In later discussions, we use the terms *reputation score* and *trust score* interchangeably.

In CommTrust, we propose an approach that combines dependency relation analysis [4], [5], a tool recently developed in natural language processing (NLP) and lexicon-based opinion mining techniques [6], [7] to extract aspect opinion expressions from feedback comments and identify their opinion orientations. We further propose an algorithm based on dependency relation analysis and Latent Dirichlet Allocation (LDA) topic modelling technique [8] to cluster aspect expressions into dimensions and com-

- *X. Zhang and L. Cui are with the School of Computer Science and IT, RMIT University, Melbourne 3001, Australia. Emails: {xiuzhen.zhang, lishan.cui}@rmit.edu.au.*
- *Y. Wang is with Macquarie University. Email: yan.wang@mq.edu.au.*

[1] http://pages.ebay.com/help/feedback/allaboutfeedback.html

pute aggregated dimension ratings and weights. We call our algorithm Lexical-LDA. Unlike conventional topic modelling formulation of unigram representations for textual documents [8], [9] our clustering is performed on the dependency relation representations of aspect opinion expressions. As a result we make use of the structures on aspect and opinion terms, as well as negation defined by dependency relations to achieve more effective clustering. To specifically address the positive bias in overall ratings, our dimension weights are computed directly by aggregating aspect opinion expressions rather than regression from overall ratings [10]–[12].

The CommTrust reputation profiles comprise dimension reputation scores and weights, as well as overall trust scores for ranking sellers. Our extensive experiments on eBay and Amazon data show that CommTrust can significantly reduce the strong positive bias in eBay and Amazon reputation systems, and solve the "all good reputation" problem and rank sellers effectively.

2 Related Work

Related work falls into three main areas: 1) computational approaches to trust, especially reputation-based trust evaluation and recent developments in fine-grained trust evaluation; 2) e-commerce feedback comments analysis and 3) aspect opinion extraction and summarisation on movie reviews, product reviews and other forms of free text.

2.1 Computational trust evaluation

The strong positive rating bias in the eBay reputation system has been well documented in literature [1]–[3], although no effective solutions have been reported. Notably in [3] it is proposed to examine feedback comments to bring seller reputation scores down to a reasonable scale, where comments that do not demonstrate explicit positive ratings are deemed negative ratings on transactions.

Similar to that buyers and sellers are referred to as individuals in e-commerce applications, terms like peers and agents are often used to refer to individuals in open systems in various applications in the trust evaluation literature. In [13] a comprehensive overview of trust models is provided. Individual level trust models are aimed to compute the reliability of peers and assist buyers in their decision making [14]–[16] whereas system level models are aimed to regulate the behaviour of peers, prevent fraudsters and ensure system security [13]. Reputation-based trust models are a class of trust models that aim to use public reputation profiles of peers to promote good behaviours and ensure security and reliability of open systems [1], [13]–[15], [17]–[22], and have been widely used in e-commerce systems [23], peer-to-peer networks [22], and multi-agent systems [13], [24].

Rating aggregation algorithms for computing individual reputation scores include simple positive feedback percentage or average of star ratings as in the eBay and Amazon reputation systems [23], the Beta reputation based on statistical distribution assumption for ratings [25], as well as more advanced models like Kalman inference [20], which also computes trust score variance and confidence level. More sophisticated reputation models consider factors like time, where recent feedback ratings carry more weights [16], [24]. PeerTrust [21], [22] is a framework for peer-to-peer systems where contextual factors are considered for computing trust scores and weights for peers. The EigenTrust algorithm [18] uses a rating matrix representation for local trust scores and computes the global ratings for peers from the rating matrix. All the above discussed models assume that feedback ratings are readily available and focus on aggregation algorithms. A couple of studies focus on gathering ratings through social networks [14], [15]. Nevertherless ratings are assumed available rather than obtained via data mining.

The multi-dimensional approach to fine-grained trust computation has been studied in agent technologies [16], [26], [27]. In [16], individual, social and ontological reputations are computed and their ratings are combined to form an overall score. In [26] the dimension scores are computed from direct experience of individual agents, and then aggregated by weighted summation. Reece et al. [27] presented a probabilistic approach considering the correlation among dimension during aggregation. In all these trust models however, weightings for dimension trust are either not considered or assumed given.

Other approaches to fine-grained trust computation have also been proposed in literature [19], [28]–[31], where specific factors for individual and transaction contexts are considered. However, many factors considered in these models are not readily available in e-commerce applications.

2.2 Feedback comment analysis

There have been studies on analysing feedback comments in e-commerce applications [3], [10], [32], [33], albeit comprehensive trust evaluation is not their focus. [3] and [32] focus on sentiment classification of feedback comments. It is demonstrated that feedback comments are noisy and therefore analysing them is a challenging problem. In [3] missing aspect comments are deemed negative and models built from aspect ratings are used to classify comments into positive or negative. In [33] a technique for summarising feedback comments is presented, aiming to filter out courteous comments that do not provide real feedback. Lu. et al [10] focuses on generating "rated aspect summary" from eBay feedback comments. Their statistical generative model is based regression on the overall transaction ratings

TABLE 1
Some sample comments on eBay

No	Comment	eBay rating
c_1	beautiful item! highly recommend using this seller!	1
c_2	bad communication, will not buy from again. super slow ship(ping). item as described.	1
c_3	quick response	1
c_4	looks good, nice product, slow delivery though.	1
c_5	top seller. many thanks. A+	1
c_6	great price and awesome service! thank you!	1
c_7	product arrived swiftly! great seller.	1
c_8	great item. best seller of ebay	1
c_9	slow postage, didn't have the product asked for, but seller was friendly.	1
c_{10}	wrong color was sent, item was damaged, did not even fit phone.	1

2.3 Aspect opinion extraction and summarisation

Our work is related to opinion mining, or sentiment analysis on free text documents. A comprehensive overview of the field is presented in [6], [7]. There has been existing work on aspect opinion mining on product reviews and movie reviews [34]–[36]. In [34] frequent nouns and noun phrases are considered aspects for product reviews, and an opinion lexicon is developed to identify opinion orientations. In [35] it is further proposed to apply lexical knowledge patterns to improve the aspect extraction accuracy. In [36] dependency relation parsing is used to mine aspect opinions for movie reviews. However these work do not group aspect opinion expressions into clusters.

Some work groups aspects into clusters, assuming aspect opinion expressions are given [37]. Recently a semi-supervised algorithm [44] was proposed to extract aspects and group them into meaningful clusters as supervised by user input seed words. Unsupervised topic modelling-based techniques have been developed to jointly model opinions and aspects (or topics), based on either the probabilistic Latent Semantic Analysis (pLSA) [9] or Latent Dirichlet Allocation (LDA) [8]. The models differ in granularities [38]–[42] and how aspects and opinions interact [38], [40], [42], [43]. All these existing work however are based on the unigram representation of documents and none of them make use of any lexical knowledge.

There has been some recent work on computing aspect ratings from overall ratings in e-commerce feedback comments or reviews [10]–[12]. Their aspect ratings and weights are computed based on regression from overall ratings and the positive bias in overall ratings is not the focus.

3 CommTrust: Comments-based Multi-dimensional trust Evaluation

We view feedback comments as a source where buyers express their opinions more honestly and openly. Our analysis of feedback comments on eBay and Amazon reveals that even if a buyer gives a positive rating for a transaction, s/he still leaves comments of mixed opinions regarding different aspects of transactions in feedback comments. Table 1 lists some sample comments, together with their rating from eBay. For example for comment c_2, a buyer gave a positive feedback rating for a transaction, but left the following comment: *"bad communication, will not buy from again. super slow ship(ping). item as described."*. Obviously the buyer has negative opinion towards the *communication* and *delivery* aspects of the transaction, despite an overall positive feedback rating towards the transaction. We call these salient aspects *dimensions* of e-commerce transactions. Comments-based trust evaluation is therefore multi-dimensional.

Definition 3.1. The overall trust score T for a seller is the weighted aggregation of dimension trust scores for the seller,

$$T = \sum_{d=1}^{m} t_d * w_d, \tag{1}$$

where t_d and w_d represent respectively the trust score and weight for dimension d ($d = 1..m$).

Following the definition of trust in by Jøsang et al. [17], the trust score on a dimension for a seller is the probability that buyers expect the seller to carry out transactions on this dimension satisfactorily. The trust score for a dimension can be estimated from the number of observed positive and negative ratings towards the dimension. Let S=$\{X_1, ..., X_n\}$ be n observations of binary positive and negative ratings, where y observations are positive ratings. S follows binomial distribution $B(n, p)$. Following the Bayes rule, p can be estimated from observations and some prior probability assumption. Assuming the Beta distribution for the prior,

$$Beta(p|\alpha, \beta) == \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

where α and β are hyper-parameters expressing prior beliefs, the Bayes estimate of p is formed by linearly combining the mean $\alpha/(\alpha+\beta)$ from prior distribution and the mean y/n, as below [45], [46]

$$\hat{p} = \frac{y+\alpha}{n+\alpha+\beta}. \tag{2}$$

Note that the Beta distribution is a special case of the Dirichlet distribution for two dimensions [46].

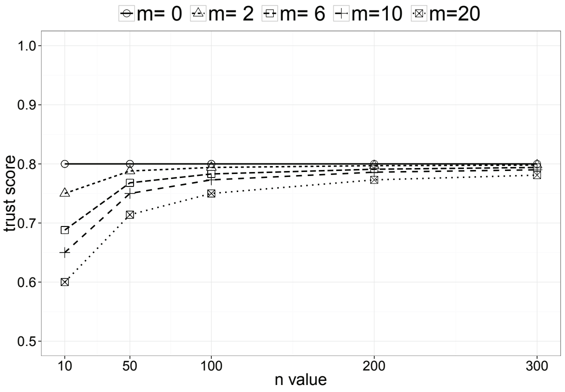


Fig. 1. The dimension trust score model

It has been shown in the Beta reputation system [25] that the assumption of Beta distribution for the prior belief leads to reasonable trust evaluation. The Beta reputation system adopts constant settings of $\alpha = \beta = 1$ for Equation 2. We develop the approach further by introducing hyper-parameter settings for α and β to suit for a varying number of observed positive and negative ratings. It is preferable to have only one parameter for trust evaluation [25]. With the prior belief of neutral tendency for trust, it can be assumed that $\alpha = \beta$. Let $\alpha + \beta = m$, then $\alpha = \beta = 1/2 * m$. The trust score for a dimension is thus defined as follows:

Definition 3.2. Given n positive (+1) and negative (-1) ratings towards dimension d, $n = |\{v_d | v_d = +1 \vee v_d = -1\}|$, the trust score for d is:

$$t_d = \frac{|\{v_d = +1\}| + 1/2 * m}{n + m}. \quad (3)$$

Equation 3 is also called *m-estimate* [47]. According to Definition 3.2, t_d is in the range of [0..1], and 0.5 represents the neutral tendency for trust. In Equation 3, m is a hyper-parameter and can be seen as pseudo counts – $1/2 * m$ counts for the positive and negative classes respectively. The higher the value of m, the more actual observations are needed to revise the natural neutral trust score of 0.5. More importantly by introducing the prior distribution using the super-parameter m, the adjustment can reduce the positive bias in ratings, especially when there are a limited number of positive and negative ratings [1], [2].

Fig. 1 plots trust score t_d by Equation 3 in relation to different settings of total number of ratings n and pseudo counts m. The figure is plotted for $y/n = 0.8$, and similar trends are observed for other values of y/n. It shows that when the total number of observed ratings n is large ($n \geq 300$), t_d is not very sensitive to the settings of m and converges to the observed positive rating frequency of 0.8. When there is a limited number of observed ratings, that is $n < 300$, an observed high positive rating frequency y/n is very likely an overestimation, and so m is set to regulate the estimated value for t_d. With $m = 2$, when $n \geq 50$ $t_d \approx 0.8$. On the other hand, with $m = 20$, only when $n \approx 300$ $t_d \approx 0.8$. From our experiments, settings of $m = 6..20$ typically give stable results. By default, we set $m = 6$.

Fig. 2 depicts the CommTrust framework. Aspect opinion expressions, and their associated ratings (positive or negative) are first extracted from feedback comments. Dimension trust scores together with their weights are further computed by clustering aspect expressions into dimensions and aggregating the dimension ratings. The algorithms for mining feedback comments for dimension ratings and for computing dimension weights will be described in Section 4.

4 Mining Feedback Comments for Dimension Ratings and Weights

We will first describe our approach based on the typed dependency anlaysis to extracting aspect opinion expressions and identifying their associated ratings. We then propose an algorithm based on LDA for clustering dimension expressions into dimensions and computing dimension weights.

4.1 Extracting aspect expressions and rating by typed dependency analysis

The typed dependency relation representation [5] is a recent NLP tool to help understand the grammatical relationships in sentences. With typed dependency relation parsing, a sentence is represented as a set of dependency relations between pairs of words in the form of (*head, dependent*), where content words are chosen as heads, and other related words depend on the heads. Fig. 3 shows an example of analysing the comment "*Super quick shipping. Product was excellent. A great deal. ALL 5 STAR.*" using the Stanford typed dependency relation parser. The comment comprises four sentences, and the sentence "*Super quick shipping.*" is represented as three dependency relations. *shipping* does not depend on any other words and is at the root level. The adjective modifier relations *amod (shipping-3, super-1)* and *amod (shipping-3, quick-2)* indicate that *super* modifies *shipping* and *quick* modifies *shipping*. The number following each word (e.g., shipping-3) indicates the position of this word in a sentence. Words are also annotated with their POS tags such as noun(NN), verb (VB), adjective (JJ) and adverb (RB).

If a comment expresses opinion towards dimensions then the dimension words and the opinion words should form some dependency relations. It has been reported that phrases formed by adjectives and nouns, and verbs and adverbs express subjectivity [48]. Among the dependency relations expressing grammatical relationships, we select the relations that express the modifying relation between adjectives and

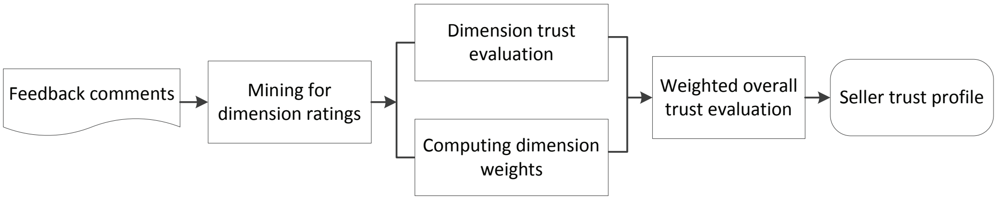
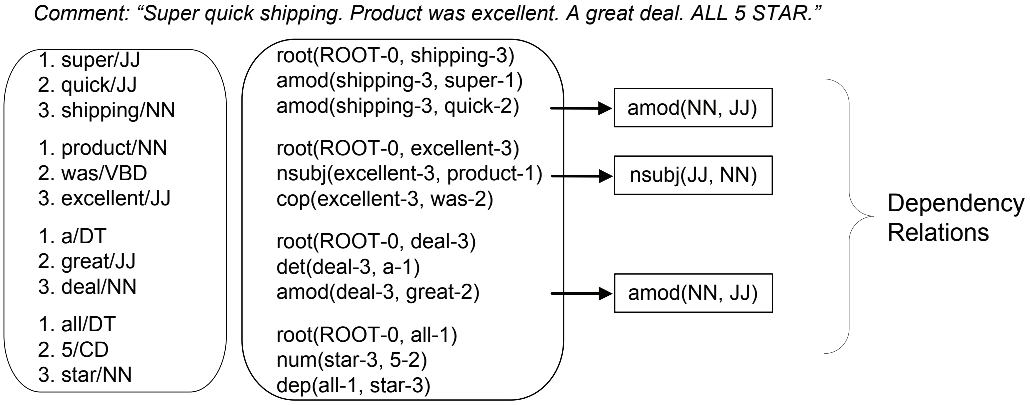

Feedback comments
Mining for dimension ratings
Dimension trust evaluation
Computing dimension weights
Weighted overall trust evaluation
Seller trust profile

Fig. 2. The CommTrust framework



NN: noun; JJ: adjective; VBD: verb past; DT: determiner; CD: cardinal number.

Fig. 3. Typed dependency relation analysis

nouns, and adverbs and verbs, as determined by the dependency relation parser. These modifying relations are listed in Table 2. It can be seen that with the modifying relations generally the noun or verb expresses the target concept under consideration whereas the adjective or adverb expresses opinion towards the target concept. The modifying relations thus can be denoted as *(modifier, head)* pairs. With the example comment in Fig. 3, the dependency relations adjective modifier *amod (NN, JJ)* and normal subject *nsubj (JJ, NN)* suggest the (modifier, head) pairs including *(super, shipping)*, *(quick, shipping)*, *(excellent, product)* and *(great, deal)*. We call these (modifier, head) pairs *dimension expressions*.

Ratings from dimension expressions towards the head terms are identified by identifying the prior polarity of the modifier terms by SentiWordNet, a public opinion lexicon. The prior polarities of terms in SentiWordNet include positive, negative or neutral, which corresponds to the ratings of +1, -1 and 0. Negations of dimension expressions are identified by the *Neg()* relation of the dependency relation parser. When a negation relation is detected the prior polarity of the modifier term is inverted.

TABLE 2
Dependency relations for dimension expressions

Dependency relation pattern	**example**
adjective modifier: *amod(NN, JJ)*	Super *quick shipping*.
adverbial modifier: *advmod(VB, RB)*	Great dealer *fast shipping*
nominal subject: *nsubj(JJ, NN)*	*Product* was *excellent*
adjectival complement: *comp(VB, JJ)*	Great CD, *arrived quick*.

NN: noun, VB: verb, JJ: adjective, and RB: adverb.
Dimension expressions are highlighted.

4.2 Clustering dimension expressions into dimensions

We propose the Lexical-LDA algorithm to cluster aspect expressions into semantically coherent categories, which we call *dimensions*. Different from the conventional topic modelling approach, which takes the document by term matrix as input, Lexical-LDA makes use of shallow lexical knowledge of dependency relations for topic modelling to achieve more effective clustering.

We make use of two types of lexical knowledge to "supervise" clustering dimension expressions into dimensions so as to produce meaningful clusters.

- Comments are short and therefore co-occurrence of head terms in comments is not very informative. We instead use the co-occurrence of dimension expressions with respect to a same modifier across comments, which potentially can provide more meaningful contexts for dimension expressions.
- We observe that it is very rare that the same aspect of e-commerce transactions is commented more than once in the same feedback comment. In other words, it is very unlikely that the dimensions expressions extracted from the same comment are about the same topic.

With the shallow lexical knowledge of dependency relation representation for dimension expressions, the clustering problem is formulated under topic modelling as follows: The dimension expressions for a same modifier term or negation of a modifier term are generated by a distribution of topics, and each topic is generated in turn by a distribution of head terms. This formulation allows us to make use of the structured dependency relation representations from the dependency relation parser for clustering. Input to Lexical-LDA are dependency relations for dimension expressions in the form of *(modifier, head)* pairs or their negations, like *(fast, shipping)* or *(not-good, seller)*.

Gibbs sampling has been proposed as approximate inference for LDA [49]. A detailed description of the derivation process for a Gibbs sampler for LDA is given in [46], while we only present the results below. Let M, K and V denote respectively the number of documents, the number of topics and the number of word tokens in the vocabulary. Let also that $\vec{\alpha}$ and $\vec{\beta}$ respectively be the hyper-parameters on the mixing proportions for topics and on the mixture components of topics. Equation 4 below is the update equation for computing the full conditional distribution of a word token w_i for a topic k, where $i = (m, n)$ denote the n^{th} word in the m^{th} document, $\vec{w} = \{w_i = t, \vec{w}_{\neg i}\}$, $\vec{z} = \{z_i = k, \vec{z}_{\neg i}\}$ and $n^{(\cdot)}_{\cdot,\neg i}$ denote counts, token i is excluded from the corresponding document or topic, and the hyper-parameters are omitted.

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) \propto \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V}(n^{(t)}_{k,\neg i} + \beta_t)} \cdot \frac{n^{(k)}_{m,\neg i} + \alpha_k}{\sum_{k=1}^{K}(n^{(k)}_{m,\neg i} + \alpha_k)} \tag{4}$$

The second type of lexical knowledge that generally two head terms from the same comment are for different dimensions is applied in LDA as a weight factor for adjusting the conditional probability for assigning head terms for a modifier term to dimensions. Specifically, for a head term w_i with index $i = (m, n)$, in computing the conditional probability for assigning w_i to topic k, we consider the evidence as presented by the head terms appearing in a same comment as w_i: when computing the conditional probability of $p(z_i = k|\vec{z}_{\neg i}, \vec{w})$, head terms in a same document with w_i and is associated with a topic other than k casts a positive vote for the conditional probability as expressed in Equation 4 and otherwise a negative vote. The weight factor is thus defined as:

$$f(z_i = k) = \frac{n^{(c,\neg k)}_{m,\neg i} - n^{(c,k)}_{m,\neg i}}{n^{(c)}_{m,\neg i}}$$

where c denotes the set of comments that w_i appears, $n^{(c,\neg k)}_{m,\neg i}$ denotes the count of head terms of m other than w_i that appear in any comment of c and is assigned to a topic other than k, $n^{(c,k)}_{m,\neg i}$ denotes the count of head terms for m other than w_i that appears in any one comment of c and is assigned to topic k, and $n^{(c)}_{m,\neg i}$ denotes the count of head terms for m other than w_i that appear in any comment of c. As a result $f(z_i = k) \in [-1, 1]$, and

$$\begin{cases} > 0 \text{ more positives votes,} \\ = 0 \text{ same number of positive and negative votes,} \\ < 0 \text{ more negative votes.} \end{cases}$$

We apply the weight factor to adjust the the computation of conditional probability in Equation 4. Given head term w_i with index $i = (m, n)$ – the n^{th} head term for a modifier term m, if there are head terms that appear in the same comment as w_i, Equation 4 is adjusted as follows:

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) \propto$$
$$(1+\alpha * f(z_i = k)) \cdot \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V}(n^{(t)}_{k,\neg i} + \beta_t)} \cdot \frac{n^{(k)}_{m,\neg i} + \alpha_k}{\sum_{k=1}^{K}(n^{(k)}_{m,\neg i} + \alpha_k)} \tag{5}$$

Three cases need to be distinguished when applying Equation 5.

- If $f(z_i = k) > 0$, that is there are more head terms in the same comments that support assigning w_i to topic k, the conditional probability estimate by the original Gibbs sampler is increased.
- If $f(z_i = k) < 0$, that is there are more head terms in the same comments that are against the assignment of w_i to topic k, the conditional probability estimate by the original Gibbs sampler is decreased.
- Otherwise $f(z_i = k) = 0$, the original Gibbs sampler estimate is kept.

In Equation 5, $\alpha \in [0, 1]$ is a parameter indicating the level of strength of the knowledge encoded in $f(z_i = k)$. The reason is that such knowledge is probabilistic in nature. The adjustment component $(1 + \alpha * f(z_i = k))$ is in the range $[1 - \alpha, 1 + \alpha]$. Note that the adjusted probability computed by Equation 5 shall be normalised for all topics afterwards.

The *(modifier, head)* structures are first used for topic discovery in [10]. In [10] Probabilistic Latent Semantic Aanalysis (PLSA) is applied where mixing weight for themes (dimensions) are assumed and optimised using the EM procedure. In our formulation the LDA model is used. More importantly we apply further lexical knowledge to constrain the process of clustering head terms to produce more meaningful clusters.

Our application of the second type of lexical knowledge to "supervise" the topic modelling process is motivated by the notion of "cannot links" in [37], although conventional LDA on documents of word tokens is applied there. Their application of constraints at the sentence level potentially can result in a large number of such constraints. In addition to the "cannot-link" constraints, "must-link" constraints are used to state that some phrases with common words

TABLE 3
The eBay dataset

Seller	Category	#comments	Feedback score	Pos feedback (%)	Detailed Seller Ratings (#ratings)			
					Item	Comm	Shipping	Cost
Seller 1	baby	5876	5481	99.6%	4.8 (2691)	4.9 (2679)	4.9 (2687)	4.8 (2660)
Seller 2	baby	4542	3618	100%	5 (221)	4.9 (223)	4.8 (223)	4.9 (229)
Seller 3	camera	2717	2609	99.4%	4.9 (832)	4.9 (829)	4.9 (837)	5 (919)
Seller 4	camera	27887	26487	99.4%	4.9 (12034)	4.9 (13046)	4.9 (12653)	5 (14019)
Seller 5	computer	5596	5457	99.9%	5 (4803)	4.9 (4998)	4.9 (4795)	5 (5299)
Seller 6	computer	27969	24199	99.9%	4.9 (15505)	4.9 (15934)	4.9 (15438)	5 (17679)
Seller 7	jewelry	3628	3194	100%	4.9 (925)	5 (986)	5 (961)	4.9 (920)
Seller 8	jewelry	60000	53624	99.7%	4.9 (44095)	5 (47734)	4.9 (45622)	5 (48088)
Seller 9	phone	34582	33237	99.4%	4.9 (3983)	5 (4375)	4.9 (4402)	5 (4717)
Seller 10	phone	29082	27392	99.5%	4.9 (5940)	5 (6507)	4.9 (6453)	5 (6929)

likely belong to the same topic. For example "battery power" and "battery life" likely belong to the same topic. Although such phrases may be widespread in product reviews, they are rare in e-commerce feedback comments. It is wroth noting that it is shown in [37] that the cannot-link constraints produce more effectiveness on the clustering results than the must-link constraints.

When *(modifier, head)* pairs and their negations are clustered into dimensions, we compute weights for dimensions. Intuitively the weight for a dimension is proportional to the total number of positive and negative ratings on the dimension. Specifically we compute the total number of *(modifier, head)* dimension expressions for the dimension. Indeed only frequent dimension expressions with head terms appearing in at least 0.1% of comments are included. The total number of dimension expressions for dimensions are normalised to produce the dimension weights.

5 Experiments

Extensive experiments on two e-commerce datasets and one hotel review datasets were conducted to evaluate various aspects of CommTrust, including the trust model and the the Lexical-LDA algorithm for clustering dimension expressions. The hotel review dataset is specifically used to demonstrate the generality of Lexical-LDA in domains other than e-commerce.

5.1 Datasets

180,788 feedback comments were crawled for ten eBay sellers on ebay.com, where two sellers were randomly selected for each of five categories on the "Shop by category" list on eBay.com, including *Cameras & Photography*, *Computers & Tablets*, *Mobile Phones & Accessories*, *Baby*, and *Jewellery & Watches*. Note that the sellers also sell products in other categories in addition to the listed categories. For evaluation of our trust model, the feedback profile for each seller were also extracted [2]:

- *The feedback score* is the total number of positive ratings for a seller from past transactions.

TABLE 4
The Amazon dataset

Seller	Category	#comments	Avg. rating
Seller 1	Electronics-Computer	4365	4.8
Seller 2	Electronics-Computer	4786	4.8
Seller 3	Electronics-Camera	3202	4.9
Seller 4	Electronics-Camera	8000	4.8
Seller 5	Electronics-Phone	5097	4.7
Seller 6	Electronics-Phone	2631	4.8
Seller 7	Jewelry-Ring	6281	4.6
Seller 8	Jewelry-Ring	1295	4.5
Seller 9	Baby-Tub	3860	4.8
Seller 10	Baby-Diaper	927	4.7

- *The positive feedback percentage* is calculated based on the total number of positive and negative feedback ratings for transactions in the last 12 months, that is $\frac{\#\text{positive-ratings}}{\#\text{positive-ratings}+\#\text{negative-ratings}}$.
- *The Detailed seller ratings* of a seller are five-star ratings on the following four aspects: *Item as described (Item)*, *Communication (Comm)*, *Shipping time (Shipping)* and *Shipping and handling charges (Cost)*. The DSR profile shows a sellers average rating and the number of ratingss. Average ratings are computed on a rolling 12-month basis, and will only appear when at least ten ratings have been received.

Details of the dataset are as shown in Table 3.

On Amazon, for a third-party seller, an average rating in the past 12 months is displayed, together with the total number of ratings. Each rating is associated with a short comment. 40,444 comments for ten third-party sellers with a large number of ratings were crawled from five categories, including *Electronics-Computer*, *Electronics-Camera*, *Electronics-Phone Jewelry-Ring*, and *Baby-Tub* and *Baby-Diaper*. Note that these sellers also sell products in other categories. A summary of the Amazon dataset is as shown in Table 4.

As shown in Tables **??** and **??**, the strong positive bias is clearly demonstrated on the eBay and Amazon datasets. On the eBay dataset, the positive feedback percentage as well as DSR five-star rating scores have little dispersion and can hardly be used by itself to rank sellers. Similarly on the Amazon dataset, the average ratings for six sellers are 4.8 or 4.9.

[2] pages.ebay.com/services/forum/feedback.html.

The TripAdvisor dataset is taken from `http://sifaka.cs.uiuc.edu/∼wang296/Data/index.html`, which was originally used in [11] and [12]. The dataset contains hotel reviews, as well as overall ratings and ratings on seven pre-defined aspects in each review. This dataset was mainly used to evaluate the applicability of Lexical-LDA for dimension clustering in domains other than e-commerce. 246,399 reviews were in the original dataset and the following preprocessing was applied: Reviews with any missing aspect rating or with less than 50 words were removed so that all reviews have coverage of all aspects. Reviews that Stanford parser can not parse were also removed. After pre-processing we have a total of 52,805 reviews.

5.2 Evaluation metrics

The ultimate goal of trust evaluation for e-commerce applications is to rank sellers and help users select trustworthy sellers to transact with. In this respect, in addition to absolute trust scores, relative rankings are more important for evaluating the performance of different trust models. To this end, we employ Kendall's τ [50] to measure the correlation between two rankings based on the number of pairwise swaps that is needed to transform one ranking into another. τ falls in $[-1, 1]$, a positive value indicates positive correlation, zero represents independence and a negative value indicates negative correlation. τ is the standard metric for comparing information retrieval systems, and it is generally considered that $\tau \geq 0.9$ for a correlation test suggests two system rankings are equivalent. A large value for $|\tau|$ with $p \leq 0.05$ suggests that two rankings are correlated, and a small value for $|\tau|$ with $p > 0.05$ suggests that two rankings are generally independent.

We employ metrics Rand index (RI) [37] and *Clustering Accuracy* (Acc) [10] to evaluate the performance of dimension clustering algorithms. RI measures both within-cluster and between-cluster agreement of two clustering algorithms. Given a pair of head terms $x \in V$ and $y \in V$, let $h(x, y)$ and $l(x, y)$ denote respectively the decision by H and L on whether x and y should be clustered into the same cluster.

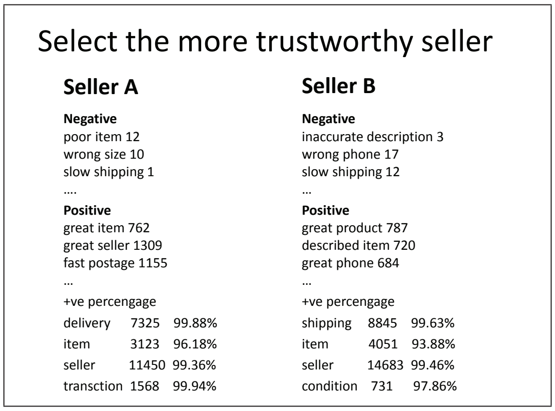
$$RI(H, L) = \frac{\sum_{x \in V} \sum_{y \in V} \theta(h(x, y), l(x, y))}{|V| \times (|V| - 1)/2}$$

where

$$\theta(h(x, y), l(x, y)) \quad = \quad \begin{cases} 1 \text{ if } h(x, y) \equiv l(x, y); \\ 0 \text{ otherwise} . \end{cases}$$

Acc measures the level of consistency between clusters produced by a clustering algorithm and the clusters by human annotation. Given a set of head terms V, consider a clustering by algorithm H and clustering by human annotation L. Each cluster $C_i (i = 1..k)$ of H is mapped to the cluster of L with the largest number of matching head terms. Let N_i denote the number of head terms in C_i with a matching head term in its corresponding cluster in L. The Acc of H is defined as

$$Acc(H) = \frac{\sum_i^k N_i}{|V|}$$



Select the more trustworthy seller

Seller A

Negative
poor item 12
wrong size 10
slow shipping 1
….

Positive
great item 762
great seller 1309
fast postage 1155
…

+ve percengage
delivery 7325 99.88%
item 3123 96.18%
seller 11450 99.36%
transction 1568 99.94%

Seller B

Negative
inaccurate description 3
wrong phone 17
slow shipping 12
…

Positive
great product 787
described item 720
great phone 684
…

+ve percengage
shipping 8845 99.63%
item 4051 93.88%
seller 14683 99.46%
condition 731 97.86%

Fig. 4. A sample pairwise preference experiment page

5.3 A user study

A user study was conducted to elicit users ranking of sellers from reading feedback comments, which was also used as the ground truth for evaluating the CommTrust multi-dimensional trust evaluation model. Inspired by evaluation techniques from the Information Retrieval community [51], experiment participants are asked to judge differences rather than make absolute ratings. For ten sellers, each seller is paired with every other seller and form 45 pairs. The orders for pairs and for sellers within pairs were randomised to avoid any presentational bias. Each pair was judged by five users and a seller preferred by at least three users was seen as a vote for the seller. The total number of preference votes from 45 pairs for each seller were used as the preference score to rank sellers.

It is infeasible to ask participants to read all comments for two sellers and choose a preferred seller. We therefore generated summaries of comments for sellers. The comment summaries for each pair of users were presented side by side to elicit users preference judgements. For a seller, we generated opinionated phrases for four dimensions, where positive and negative phrases for each dimension are ordered by decreasing frequency. The three most frequent positive and negative phrases for each dimension formed the summary for a seller. An example page for the survey is shown in Figure 4.

Results from the experiment for eBay and Amazon sellers are summarised in Table 5. Under the column heading of Comment rank is the ranking of sellers by user preferences after participants read the comment

TABLE 5
Seller rankings by reading comments in user studies

eBay seller	eBay rank	Comment rank	Amazon seller	Amazon rank	Comment rank
Seller 1	6	6	Seller 1	5	5
Seller 2	8	5	Seller 2	4	6
Seller 3	10	10	Seller 3	7	2
Seller 4	4	2	Seller 4	1	1
Seller 5	7	3	Seller 5	3	8
Seller 6	5	1	Seller 6	8	7
Seller 7	9	7	Seller 7	2	4
Seller 8	1	9	Seller 8	9	9
Seller 9	2	4	Seller 9	6	3
Seller 10	3	8	Seller 10	10	10
Kendall's τ=0.1111, p-value=0.7275			Kendall's τ=0.4222, p-value= 0.1083		
rank-diff=3			rank-diff=1.8		

summaries for sellers. The correlation between rankings are measured by Kendall's τ. The rank difference between two ranking vectors is defined as:

$$\text{rank-diff} = \frac{\sum_i rank(i) - rank'(i)}{N}$$

where rank(i) and rank'(i) are respectively the rank for seller i by two ranking methods, and N=10. The low Kendall's τ value (0.1111 and 0.4222) and high p-value (0.7275 and 0.1083) suggest that on eBay and Amazon, user preference rankings after reading comment summaries are not strongly correlated with the rankings by the respective eBay and Amazon reputation systems. This suggests that the comments contain distinct information for users to rank sellers. The ranking difference of 3 for ten eBay users between rankings by reading comments and by eBay reputation system suggests that on average there is a difference of 3 ranks for sellers by the two approaches. Similarly for Amazon sellers there is difference of 1.8 ranks on average. Our user study demonstrates that it can be speculated that content of comments can be used to reliably evaluate the trustworthiness of sellers, which is the objective of CommTrust.

5.4 Evaluation of the trust model

Table 6 lists the CommTrust overall trust scores for ten eBay sellers and ten Amazon sellers for 4,7 and 10 dimensions respectively. As the ground truth, the rankings by reading comment summaries for sellers are also listed (under the heading Comm rank). For both eBay and Amazon sellers, on all 4, 7 and 10 dimensions, the rankings by CommTrust are strongly correlated with the ground truth rankings, as demonstrated by the high Kendall's τ and low p-values (less than 0.05). The negative Kendall's τ values are a result of that the ranks are inversely ordered in relation to trust scores. This is suggesting that CommTrust has computed the dimension ratings from comments and they match users' preferences after reading the comments. The number of dimensions does not affect how well the trust scores are correlated with the user rankings.

A strength of CommTrust is that the relative weights that users have placed on different dimensions in their feedback comments can be inferred. However, it is hard to elicit the weights from users when they write the feedback comments. We therefore evaluate our dimension weight prediction indirectly. To verify the effectiveness of the dimension weights in the overall trust score, we compute the unweighted overall trust scores for sellers, and compare the ranking of sellers by unweighted overall trust scores with the ground truth ranking by users. Table 7 shows the result. It can be seen that without weightings for dimensions, the trust scores for sellers are not correlated with the ground truth ranking of sellers, as demonstrated by low Kendall's τ with all p-value greater than 0.05. This result holds for eBay and Amazon sellers, and all 4, 7 and 10 dimensions.

The dimension trust scores and weights together form the dimensional trust profiles for sellers. The dimensional trust profiles for ten eBay sellers for four dimensions are shown in Table 8. Note that the four dimensions discovered by CommTrust for a seller are the statistically important dimensions that users expressed opinions on in their feedback comments and may not necessarily correspond to the four aspects as specified by eBay DSR ratings. Nevertheless *item* and *shipping* indeed are the dimensions where users comment the most on. In Table 8 the dimensional trust score and weight for the *item* dimension has been underlined. It can be seen that users have substantially different ratings on the *item* dimension for different sellers and put on different weights.

Table 9 lists the dimensional trust profiles for ten Amazon sellers. The dimensions *item*, *shipping* and *seller (service)* are the three "hot" dimensions for feedback comments across ten sellers. The fourth dimension includes topics like *condition*, *price* or *packaging*. Generally compared with the eBay dataset, dimensional trust scores are more dispersely distributed among the ten Amazon sellers. The first two columns of Table 9 list the dimensional trust scores and weights for the *item* dimension. Obviously the ten sellers are significantly different – trust scores vary from 0.6545

TABLE 6
Overall trust scores by CommTrust for ten eBay sellers and ten Amazon sellers

eBay					Amazon				
eBay seller	Comm rank	4 dims	7 dims	10 dims	Amazon seller	Comm rank	4 dims	7 dims	10 dims
Seller 1	6	0.9798	0.9777	0.9766	Seller 1	5	0.8876	0.8887	0.8861
Seller 2	5	0.9865	0.9848	0.9828	Seller 2	6	0.8924	0.8957	0.8945
Seller 3	10	0.9771	0.9741	0.9700	Seller 3	2	0.9259	0.9223	0.9201
Seller 4	2	0.9837	0.9836	0.9824	Seller 4	1	0.8896	0.8875	0.8837
Seller 5	3	0.9852	0.9824	0.9824	Seller 5	8	0.8750	0.8718	0.8597
Seller 6	1	0.9850	0.9855	0.9851	Seller 6	7	0.8899	0.8857	0.8834
Seller 7	7	0.9798	0.9783	0.9743	Seller 7	4	0.8787	0.8832	0.8791
Seller 8	9	0.9717	0.9732	0.9725	Seller 8	9	0.8643	0.8573	0.8516
Seller 9	4	0.9823	0.9818	0.9805	Seller 9	3	0.9360	0.9317	0.9302
Seller 10	8	0.9807	0.9814	0.9819	Seller 10	10	0.7855	0.7871	0.7961
Kendall's τ		-0.6000	-0.6889	-0.7333	Kendall's τ		-0.5556	-0.6000	-0.6000
p-value		0.0167	0.0047	0.0022	p-value		0.0286	0.0167	0.0167

TABLE 7
Unweighted overall trust scores for ten eBay sellers and ten Amazon sellers

eBay					Amazon				
eBay seller	Comm rank	4 dims	7 dims	10 dims	Amazon seller	Comm rank	4 dims	7 dims	10 dims
Seller 1	6	0.9785	0.9433	0.8712	Seller 1	5	0.8543	0.8488	0.8194
Seller 2	5	0.9811	0.9039	0.9280	Seller 2	6	0.8959	0.8673	0.8557
Seller 3	10	0.9224	0.9563	0.8880	Seller 3	2	0.8883	0.8252	0.8021
Seller 4	2	0.9794	0.9728	0.9655	Seller 4	1	0.8563	0.8229	0.8228
Seller 5	3	0.9243	0.9253	0.9178	Seller 5	8	0.8410	0.8283	0.7665
Seller 6	1	0.9820	0.9042	0.9521	Seller 6	7	0.8691	0.7866	0.8063
Seller 7	7	0.9819	0.9492	0.9296	Seller 7	4	0.8971	0.8233	0.8233
Seller 8	9	0.9690	0.9535	0.9416	Seller 8	9	0.8579	0.8168	0.8125
Seller 9	4	0.9571	0.9618	0.9204	Seller 9	3	0.8865	0.8361	0.8153
Seller 10	8	0.9691	0.8976	0.9689	Seller 10	10	0.7387	0.6949	0.6456
Kendall's τ		-0.3333	0.0667	-0.0667	Kendall's τ		-0.3778	-0.2000	-0.3333
p-value		0.2164	0.8618	0.8618	p-value		0.1557	0.4843	0.2164

TABLE 8
The dimensional trust profiles for ten eBay sellers

Seller	Dim 1	Weight	Dim 2	Weight	Dim 3	Weight	Dim 4	Weight	Overall
seller 1	0.9950	0.3058	0.9585	0.0359	$\underline{0.9612}$	$\underline{0.2816}$	0.9835	0.3766	0.9798
seller 2	0.9825	0.2347	0.9763	0.2453	$\underline{0.9691}$	$\underline{0.0150}$	0.9937	0.5051	0.9865
seller 3	0.9864	0.3466	0.9067	0.0225	$\underline{0.9510}$	$\underline{0.1292}$	0.9806	0.5017	0.9771
seller 4	$\underline{0.9476}$	$\underline{0.1134}$	0.9912	0.0544	0.9854	0.4875	0.9921	0.3447	0.9837
seller 5	0.9891	0.5488	$\underline{0.9354}$	$\underline{0.0734}$	0.9856	0.0361	0.9895	0.3417	0.9852
seller 6	$\underline{0.9465}$	$\underline{0.1484}$	0.9881	0.4831	0.9965	0.3258	0.9956	0.0427	0.9850
seller 7	0.9803	0.1834	$\underline{0.9709}$	$\underline{0.4009}$	0.9878	0.1406	0.9883	0.2751	0.9798
seller 8	0.9944	0.2081	0.9778	0.4407	$\underline{0.9578}$	$\underline{0.3245}$	0.8632	0.0267	0.9717
seller 9	0.9944	0.3465	0.9558	0.0058	0.9894	0.5087	$\underline{0.9272}$	$\underline{0.1390}$	0.9823
seller 10	0.9726	0.0175	$\underline{0.9280}$	$\underline{0.1589}$	0.9889	0.5048	0.9945	0.3188	0.9807

Note: The dimensional trust scores and weights for the *item* dimension are underlined.

for Seller 10 to 0.9738 for Seller 7, whereas weights vary from 0.1299 for Seller 6 to 0.5294 for Seller 5.

Fig. 5 depicts the dimensional trust profiles for three eBay sellers Sellers 1, Seller 2 and Seller 3, where they have the same four dimensions, including *shipping*, *cost/response*, *item* and *seller*. For each seller, the upward bars represent trust scores for dimensions while the downward bars represent their weights. For example while having a high overall trust score of 0.9771, Seller 3 has a low dimension trust score of 0.9067 for the *response* dimension (Dimension 2). The figure clearly illustrates the variation of dimension trust for each seller horizontally and those across different sellers vertically. Such comprehensive trust profiles certainly can cater to users preferences for different dimensions and guide users in making informed decisions when choosing sellers.

5.5 Evaluation of Lexical-LDA

Informal language expressions are widely used in feedback comments. Some pre-processing was first performed: Spelling correction was applied. Informal expressions like *A+++* and *thankx* were replaced with *AAA* and *thanks*. The Stanford dependency relation parser was then applied to produce the dependency relation representation of comments and dimension expressions were extracted. The dimension expressions were then clustered to dimensions by the Lexical-LDA algorithm.

TABLE 9
The dimensional trust profiles for ten Amazon sellers

Seller	Dim 1	Weight	Dim 2	Weight	Dim 3	Weight	Dim 4	Weight	Overall
seller 1	0.6898	0.1556	0.9462	0.4754	0.9704	0.1955	0.8109	0.1736	0.8876
seller 2	0.9624	0.1884	0.9427	0.3540	0.9249	0.1897	0.7535	0.2678	0.8924
seller 3	0.8571	0.1560	0.9663	0.2008	0.9702	0.5202	0.7597	0.1230	0.9259
seller 4	0.7981	0.1388	0.7241	0.1645	0.9405	0.4960	0.9627	0.2007	0.8896
seller 5	0.9377	0.5294	0.8208	0.1509	0.9288	0.1524	0.6766	0.1674	0.8750
seller 6	0.8286	0.1299	0.9662	0.1675	0.9571	0.4789	0.7243	0.2237	0.8899
seller 7	0.9738	0.1307	0.8744	0.1030	0.9415	0.3450	0.7987	0.4213	0.8787
seller 8	0.9222	0.3535	0.9167	0.1086	0.7573	0.1371	0.8355	0.4007	0.8643
seller 9	0.9841	0.2231	0.8473	0.1555	0.7500	0.0688	0.9646	0.5526	0.9360
seller 10	0.6545	0.1439	0.9136	0.2336	0.9268	0.4103	0.4600	0.2123	0.7855

Note: The dimensional trust scores and weights for the *item* dimension are underlined.

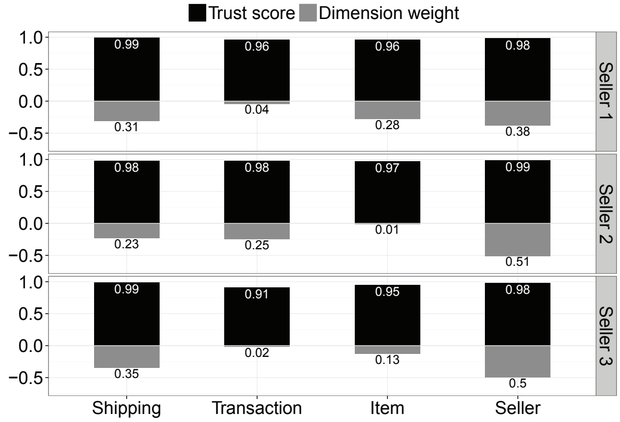


Fig. 5. The dimension trust profiles by CommTrust for sellers

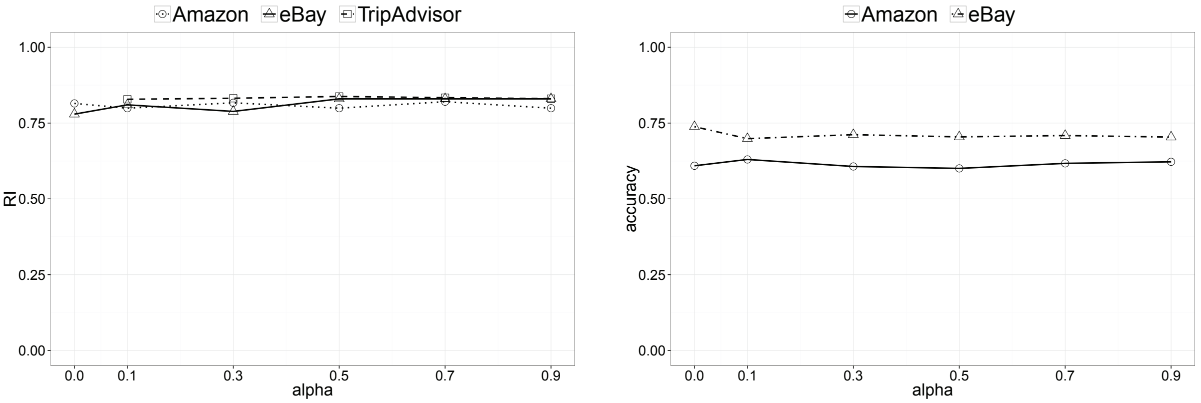
To evaluate Lexical-LDA, the ground truth for clustering was first established. Dimension expressions are *(modifier, head)* pairs, and to remove noise only those pairs with support for head terms of at least 0.1% or three comments (whichever is larger) were considered for manual clustering. Some head terms resulted from parsing errors that do not appear to be an aspect were discarded. Examples of such terms include *thanks*, *ok* and *A+++*, In the end a total of maximum 100 head terms were manually clustered based on the inductive approach to analysing qualitative data [52]. We first grouped head terms into categories according to their conceptual meaning – some head terms may belong to more than one category, and some orphan words were discarded. We then combined some categories with overlapping head terms into a broader category, until some level of agreement was reached between annotators. [3] As a result of this manual labelling process for the eBay and Amazon dataset, the feedback comments for each seller finally seven clusters are obtained.

Lexical-LDA was implemented based on the Mallet topic modelling toolkit [53]. With aspect expressions in the form of *(modifier, head)* pairs, the modifier term by head term matrix formed the input for Lexical-LDA. In constructing the cannot-link head term list for a head term (c.f. Section 4.2), only head terms appearing together with the head term in at least 0.1% of or three (whichever is larger) comments were considered. The purpose was to remove the otherwise many spurious cannot-link head terms. The Lexical-LDA parameter settings were: prior pseudo counts for topics and terms were set as $\alpha_k = 0.1$ and $\beta_t = 0.01$ (See Equation (5)), the number of topics K = 4, 7, 10 for evaluating the trust model and number of iterations was set to 1000.

We evaluate Lexical-LDA against standard LDA for clustering and against the human clustering result. As there are seven categories by human clustering, $K = 7$ for Lexical-LDA. Figure 6(a) plots the RI of Lexical-LDA at different settings of α. Note that the data point for $\alpha = 0$ corresponds to the standard LDA. In addition to the eBay and Amazon datasets, to demonstrate the generality of our approach, the performance of Lexial-LDA on the TripAdvisor dataset is also plotted. For eBay and Amazon data, each plotted data point is the average for ten sellers. On eBay data, RI of Lexical-LDA hovers over 0.78 $\sim$ 0.83, and Lexical-LDA significantly outperforms standard LDA for $\alpha > 0$ except $\alpha = 0.3$ (p-value < 0.05, paired two-tail t-test). Comparable RI is observed on TripAdvisor and Amazon datasets. Our experiment results indicate that Lexical-LDA has steady performance across different domains.

Fig. 6(b) plots the accuracy of Lexical-LDA with different settings of α. As can be seen in the graph, accuracies hover over 0.70 $\sim$ 0.74 on eBay data and 0.61 $\sim$ 0.63 on Amazon data. There are not statistically significant differences in accuracies between Lexical-LDA with $\alpha > 0$ and standard LDA, on either Amazon or eBay datasets. However clustering accuracy only measures how automatic clustering matches the human clustering, rather than the coherence within clusters by clustering algorithms. Table 10 shows the clusters of head terms for seven dimensions for eBay Seller 1 from manual clustering, Lexical-LDA ($\alpha = 0.5$) and standard LDA respectively. Each head term is grouped to the dimension with the highest

[3] Manual clustering was performed by the first two authors. Inconsistency was resolved by discussion.



(a) The RI of Lexical-LDA

(b) The Accuracy of Lexical-LDA

Fig. 6. Evaluation of Lexical-LDA dimension clustering

TABLE 10
The head term clusters for dimensions

Dim	Manual clustering	Lexical-LDA (α=0.5)	Standard LDA
1	item, bag, product, dress, earrings, outfit, top, ring, shoes, coat, necklace, jacket, stuff, one, curtains, handbag, boots, zip, toy, backpack, suit, material, goods, piece, scarf, leggings	item: 532, bag: 146, dress: 70, earrings: 49, outfit:45, coat: 16, top: 16, ring:14, one:11, shoes: 11, jacket:11, necklace: 11, tfit: 8, handbag: 7, look: 7, received: 7, goods: 6, scarf: 3, product: 3	item: 341, bag: 199, dress: 74, earrings: 61, outfit: 50, shoes: 17, coat: 16, ring: 15, necklace: 13, jacket: 11, one: 10, look: 10, curtains: 8, fit: 7, handbag: 6, suit: 6, received: 6, track: 5, toy: 3, piece: 3, leggings: 3, scarf: 3
2	quality, condition, look, size, color, description, fit, described, design	look: 16, size: 10, material: 10, curtains: 8, color: 8, zip: 6, design: 4	size: 11, refund: 8, material: 8, zip: 5, color: 5, design: 5, order: 4, business: 4, post: 3
3	delivery, shipping, postage, dispatch, time, arrived, received, post, shipment, arrival, came	delivery: 1139, payment: 179, shipping: 69, response: 59, postage: 50, dispatch: 25, despatch: 18, deal: 10, came: 10, arrival: 7, arrived: 6, shipment: 5, post: 5	delivery: 1096, shipping: 60, response: 58, postage: 45, dispatch: 22, despatch: 18, deal: 10, came: 10, arrival: 7, arrived: 6, shipment: 5
4	seller, ebayer	seller: 286, ebayer: 286, bayer: 5, described: 4, leggings: 3, track: 3	seller: 519, ebayer: 409, service: 249, communication: 149, product: 138, price: 44, quality: 39, value: 39, buy: 29, condition: 19, looks: 16, top: 15, items: 13, purchase: 13, ebay: 12, time: 11, bayer: 8, stuff: 7, described: 5, boots: 4, description: 4, backpack: 4
5	service, response, track, communication	communication: 142, service: 133, product: 106, quality: 55, price: 46, value: 40, buy: 29, condition: 28, ebay: 11, time: 10, stuff: 8, purchase: 6, boots: 5, description: 5, backpack: 5	goods: 5
6	transaction, buy, deal, purchase, order, business	transaction: 165	transaction: 160
7	payment, price, value, refund	refund: 12, order: 6, business: 5, suit: 4, toy: 4, piece: 1	payment: 147

frequency. We can see that Lexical-LDA has significantly higher within-cluster coherence than standard LDA. For example Dimension 2 is about the details of items, including for example *quality*, *condition*, *look*, *size* and *colour*. All head terms from Lexical-LDA in this dimension (arguably excluding *curtains*) are indeed about items sold by the seller, although some details are missing. In comparison, the head terms in this dimension from standard LDA are very dispersed and some are not related to items at all, including *refund*, *order*, *business*, and *post*. We believe that the supervision from non-link constraints for head terms helps to produce the meaningful clusters for head terms.

TABLE 11
The precision of identifying different ratings

	Positive	Negative	Neutral	Average
eBay	0.86±0.03	0.60±0.03	0.94±0.02	0.80±0.18
Amazon	0.94±0.03	0.68±0.11	0.93±0.02	0.85±0.15

SentiWordNet is used to decide the prior orientation of modifier terms. Table 11 lists the precision of our approach for idenfying positive, negative and neutral ratings on the eBay and Amazon datasets respectively. Precision is calculated as the proportion of correctly identified from all *(modifier, head)* pairs computed for each polarity of positive, negative and neutral. It can be seen that generally our approach achieves reasonably good average precision for all types of ratings — 0.80 ± 0.18 on eBay data and 0.85 ± 0.15 on Amazon data respectively. However the precision for the negative ratings is low, which is mainly due to that SentiWordNet is a general lexicon and as a result some word polarity annotation does not suit the e-commerce application. For example *short* is annotated as neutral and negative in SentiWordNet, and using the latter annotation leads to wrong decision for our application. The problem of adapting general opinion lexicons to different domains is an interesting problem outside the scope of this paper, and readers are referred to the relevant literature (e.g., [54], [55]).

6 Conclusions

The "*all good reputation*" problem is well known for the reputation management systems of popular e-commerce web sites like eBay and Amazon. The high reputation scores for sellers can not effectively rank sellers and therefore can not guide potential buyers to select trustworthy sellers to transact with. On the other hand, it is observed that although buyers may give high feedback ratings on transactions, they often express direct negative opinions on aspects of transactions in free text feedback comments. In this paper we have proposed to compute comprehensive multi-dimensional trust profiles for sellers by uncovering dimension ratings embedded in feedback comments. Extensive experiments on feedback comments for eBay and Amazon sellers demonstrate that our approach computes trust scores highly effective to distinguish and rank sellers.

We have proposed effective algorithms to compute dimension trust scores and dimension weights automatically via extracting aspect opinion expressions from feedback comments and clustering them into dimensions. Our approach demonstrates the novel application of combining natural language processing with opinion mining and summarisation techniques in trust evaluation for e-commerce applications.

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments which help improve the quality of this paper.

References

[1] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation Systems: Facilitating Trust in Internet Interactions," *Communications of the ACM*, vol. 43, pp. 45–48, 2000.

[2] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," *The Economics of the Internet and E-commerce*, 2002.

[3] J. O'Donovan, B. Smyth, V. Evrim, and D. McLeod, "Extracting and visualizing trust relationships from online auction feedback comments," in *Proc. IJCAI'07*, 2007, pp. 2826–2831.

[4] M. De Marneffe, B. MacCartney, and C. Manning, "Generating typed dependency parses from phrase structure parses," in *Proc. LREC*, vol. 6, 2006, pp. 449–454.

[5] M. De Marneffe and C. Manning, "The stanford typed dependencies representation," in *Proc. the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 2008.

[6] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1-2, pp. 1–135, Jan. 2008.

[7] B. Liu, *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers, 2012.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[9] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.

[10] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proc. the 18th Int. Conf. on WWW*, 2009.

[11] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis without aspect keyword supervision," in *Proc. the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 618–626.

[12] ——, "Latent aspect rating analysis on review text data: a rating regression approach," in *Proc. the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 783–792.

[13] S. Ramchurn, D. Huynh, and N. Jennings, "Trust in multi-agent systems," *The Knowledge Engineering Review*, 2004.

[14] B. Yu and M. P. Singh, "Distributed reputation management for electronic commerce," *Computational Intelligence*, 2002.

[15] M. Schillo, P. Funk, and M. Rovatsos, "Using trust for detecting deceptive agents in artificial societies," *Applied Artificial Intelligence*, vol. 14, no. 8, pp. 825–848, 2000.

[16] J. Sabater and C. Sierra, "Regret: reputation in gregarious societies," in *Proc. the fifth international conference on Autonomous agents*. ACM, 2001, pp. 194–195.

[17] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[18] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proc. the 12th Int. Conf. on WWW*, 2003.

[19] A. Rettinger, M. Nickles, and V. Tresp, "Statistical relational learning of trust," *Machine learning*, vol. 82, pp. 191–209, 2011.

[20] X. Wang, L. Liu, and J. Su, "RLM: A general model for trust representation and aggregation," *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 131–143, 2012.

[21] L. Xiong and L. Liu, "A reputation-based trust model for peer-to-peer e-commerce communities," in *Proc. IEEE Int. Conf. E-Commerce*, 2003, pp. 275–284.

[22] ——, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[23] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood, "The value of reputation on ebay: A controlled experiment," *Experimental Economics*, vol. 9, no. 2, pp. 79–101, 2006.

[24] Y. Wang and M. Singh, "Trust representation and aggregation in a distributed agent system," in *Proc. the national conference on Artificial Intelligence*, 2006.

[25] A. Jøsang and R. Ismail, "The beta reputation system," in *Proc. the 15th bled electronic commerce conference*, 2002, pp. 41–55.

[26] N. Griffiths, "Task delegation using experience-based multi-dimensional trust," in *Proc. the fourth international joint conference on AAMAS*, 2005, pp. 489–496.

[27] S. Reece, A. Rogers, S. Roberts, and N. Jennings, "Rumours and reputation: Evaluating multi-dimensional trust within a decentralised reputation system," in *Proc. the 6th international joint conference on AAMAS*, 2007, pp. 165–172.

[28] Y. Wang and E. Lim, "The evaluation of situational transaction trust in e-service environments," in *Proc. IEEE International Conference on e-Business Engineering*, 2008, pp. 265–272.

[29] Y. Zhang and Y. Fang, "A fine-grained reputation system for reliable service selection in peer-to-peer networks," *IEEE Transactions on Parallel and Distributed Systems*, 2007.

[30] H. Zhang, Y. Wang, and X. Zhang, "Efficient contextual transaction trust computation in e-commerce environments," in *Proc. the 11th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communicatios (TrustCom 2012)*, 2012.

[31] ——, "A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments," in *Proc. the fifth IEEE Int. Conf. on Service-oriented Computing and Applications (SOCA 2012)*, 2012.

[32] M. Gamon, "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis," in *Proc. the 20th Int. Conf. on Computational Linguistics*, 2004.

[33] Y. Hijikata, H. Ohno, Y. Kusumura, and S. Nishida, "Social summarization of text feedback for online auctions and interactive presentation of the summary," *Knowledge-Based Systems*, vol. 20, no. 6, pp. 527–541, 2007.

[34] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. the fourth Int. Conf. on KDD*, 2004, pp. 168–177.

[35] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Computational linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[36] L. Zhuang, F. Jing, X. Zhu, and L. Zhang, "Movie review mining and summarization," in *Proc. the 15th ACM Int. Conf. on Information and knowledge management*, 2006, pp. 43–50.

[37] Z. Zhai, B. Liu, H. Xu, and P. Jia, "Constrained LDA for grouping product features in opinion mining," in *Proc. Advances in knowledge discovery and data mining*. Springer, 2011, pp. 448–459.

[38] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proc. the 16th international conference on World Wide Web*, 2007, pp. 171–180.

[39] I. Titov and R. T. McDonald, "A joint model of text and aspect ratings for sentiment summarization." in *Proc. ACL*, 2008, pp. 308–316.

[40] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proc. the 18th ACM conference on Information and knowledge management*, 2009, pp. 375–384.

[41] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proc. the 17th international conference on World Wide Web*, 2008, pp. 111–120.

[42] S. Brody and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews," in *Proc. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 804–812.

[43] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid," in *Proc. the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 56–65.

[44] A. Mukherjee and B. Liu, "Aspect extraction through semi-supervised modeling," in *Proc. the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 2012, pp. 339–348.

[45] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Press, 1990.

[46] G. Heinrich, "Parameter estimation for text analysis," University of Leipzig, Tech. Rep., 2005.

[47] K. Karplus, "Evaluating regularizers for estimating distributions of amino acids," in *Proc. the third Int. Conf. on Intelligent Systems for Molecular Biology*, vol. 3, 1995, pp. 188–196.

[48] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proc. the 40th annual meeting on association for computational linguistics*, 2002, pp. 417–424.

[49] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *the National academy of Sciences of the United States of America*, vol. 101, no. Suppl 1, pp. 5228–5235, 2004.

[50] D. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. CRC Press, 2003.

[51] P. Thomas and D. Hawking, "Evaluation by comparing result sets in context," in *Proc. the 15th ACM international conference on Information and knowledge management*, 2006, pp. 94–101.

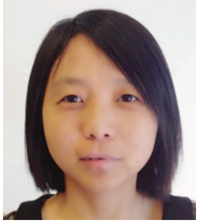
[52] D. R. Thomas, "A general inductive approach for analyzing qualitative evaluation data," *American journal of evaluation*, vol. 27, no. 2, pp. 237–246, 2006.

[53] A. K. McCallum, "MALLET: a machine learning for language toolkit," 2002. [Online]. Available: http://mallet.cs.umass.edu

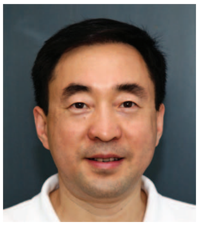
[54] A. Fahrni and M. Klenner, "Old wine or warm beer: Target-specific sentiment analysis of adjectives," in *Proc. of the Symposium on Affective Language in Human and Machine, AISB*, 2008, pp. 60–63.

[55] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *ACL*, vol. 7, 2007, pp. 440–447.

Xiuzhen Zhang is a Senior Lecturer in the School of Computer Science and IT, RMIT University, Australia. Her research interests include data mining and data analytics. She has published over 40 international journal and conference papers in the areas. She has been on the program committees of several international conferences, and a regular reviewer for several international journals.



Lishan Cui is a research student in the School of Computer Science and IT, RMIT University, Australia. Her research interests include data mining for e-commerce applications.



Yang Wang is an Associate Professor in the Department of Computing, Macquarie University, Australia. His research interests include trust computing, e-commerce and social network analysis. He has published over 70 international journal and conference papers. He serves on the editorial board of several international journals and is a senior member of the IEEE.