Exploring Linkablility of Community Reviewing

Anonymous and Unlinkable

Abstract

Large numbers of people all over the world read and contribute to various review sites. Many contributors are understandably concerned about privacy in general and, specifically, about linkability of their reviews (and accounts) across multiple review sites. In this paper, we study linkability of community-based reviewing and try to answer the question: *to what extent are "anonymous" reviews linkable, i.e., highly likely authored by the same contributor?* Based on a very large set of reviews from one very popular site (Yelp), we show that a high percentage of ostensibly anonymous reviews can be linked with very high confidence. This is despite the fact that we use very simple models and equally simple features set. Our study suggests that contributors reliably expose their identities in reviews. This has important implications for cross-referencing accounts between different review sites. Also, techniques used in our study could be adopted by review sites to give contributors feedback about privacy of their reviews.

1. Introduction

In recent years, popularity of various types of review and community-knowledge sites has substantially increased. Prominent examples include Yelp, Tripadvisor, Epinions, Wikipedia, Expedia and Netflix. They attract multitudes of readers and contributors. While the former usually greatly outnumber the latter, contributors can still number in hundreds of thousands for large sites, such as Yelp or Wikipedia. For example, Yelp had more than 39 million visitors and reached 15 million reviews in late 2010 [2]. To motivate contributors to provide more (and more useful/informative) reviews, certain sites even offer rewards [3].

Some review sites are generic (e.g., Epinions) while others are domain-oriented (e.g., Tripadvisor). Large-scale reviewing is not limited to review-oriented sites; in fact, many retail sites encourage customers to review products. e.g., Amazon and Netflix.

With the surge in popularity of community- and peer-based reviewing, more and more people contribute to review sites. At the same time, there has been an increased awareness with regard to personal privacy. Internet and Web privacy is a broad notion with numerous aspects, many of which have been explored by the research community. However, privacy in the context of review sites has not been adequately studied. Although lately there has been a significant amount of research related to reviews, its focus has been mainly on extracting and summarizing opinions from reviews [6, 9, 20] as well as determining authenticity of reviews [11, 12, 14].

In the context of community-based reviewing, contributor privacy has several aspects: (1) some review sites do not require accounts (i.e., allow ad hoc reviews) and contributors might be concerned about linkability of their reviews, and (2) many active contributors have accounts on multiple review sites and prefer these accounts not be linkable. The flip side of the privacy problem is faced by review sites themselves: how to address spam-reviews and sybil-accounts?

The goal of this paper is to explore linkability of reviews by investigating how close and related are a person's reviews. That is, how accurately we can link a set of anonymous reviews to their original author. Our study is based on over $1,000,000$ reviews and $\simeq \ 2,000$ contributors from Yelp. Our results clearly illustrate that most (up to 99% in some cases) reviews by relatively active/frequent contributors are highly linkable. This is despite the fact that our approach is based on simple models and simple feature sets. For example, using only alphabetical letter distributions, we can link up to 83% of anonymous reviews. We anticipate two contributions of this work: (1) extensive assessment of reviews' linkability, and (2) several models that quite accurately link "anonymous" reviews.

Our results have several implications. One of them is the ability to cross-reference contributor accounts between multiple review sites. If a person regularly contributes to two review sites under different accounts, anyone can easily link them, since most people tend to consistently maintain their traits in writing reviews. This is possibly quite detrimental to personal privacy. Another implication is the ability to correlate reviews ostensibly emanating from different accounts

that are produced by the same author. Our approach can thus be very useful in detecting self-reviewing and, more generally, review spam [11] whereby one person contributes from multiple accounts to artificially promote or criticize products or services.

One envisaged application of our technique is to have it integrated into review site software. This way, review authors could obtain real-time feedback indicating the degree of linkability of their reviews. It would then be up to each author to adjust (or not) the writing style and other characteristics.

Organization: Section 2 provides background information about techniques used in our experiments. The sample dataset is described in Section 3 and study settings are addressed in Section 4. Next, our analysis methodology is presented in Section 5. Section 6 discusses issues stemming from this work and Section 7 sketches out some directions for the future. Then, Section 8 overviews related work and Section9 concludes the paper.

2. Background

This section provides a bit of background information about statistical tools used in the study reported on in this paper. We use two well-known approaches based on: (1) Naïve Bayes Model [13], (2) Kullback-Leibler Divergence Metric [5]. We briefly describe them below.

2.1 Naïve Bayes Model

Naïve Bayes Model (NB) is a probabilistic model based on the eponymous assumption stating that all features/tokens are conditionally independent given the class. Given the tokens $T_1, T_2, ..., T_n$ in a document D, we compute the conditional probability of a document class C as follows:

$$P(C|D) = P(C|T_1, T_2, ..., T_n) = \frac{P(T_1, T_2, ..., T_n|C)P(C)}{P(T_1, T_2, ..., T_n)}$$

According to the Naïve Bayes assumption,

$$P(T_1, T_2, ..., T_n|C) = P(T_1|C)P(T_2|C).....P(T_n|C)$$

Therefore,

$$P(C|T_1, T_2, ..., T_n) = \frac{P(T_1|C)P(T_2|C).....P(T_n|C)P(C)}{P(T_1, T_2, ..., T_n)}$$

To use NB for classification, we return the class value with maximum probability:

$$Class = argmax_C P(C|D) = argmax_C P(C|T_1, T_2, ..., T_n) \quad (1)$$

Since $P(T_1, T_2, ..., T_n)$ is the same for all C values, and assuming $P(C)$ is the same for all class values, the above equation is reduced to:

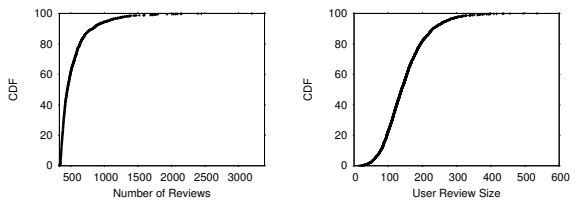
$$Class = argmax_C P(T_1|C)P(T_2|C).....P(T_n|C)$$

Probabilities are estimated using the Maximum-Likelihood estimator [5] as follows:

$$P(T_i|C) = \frac{Num\ of\ Occurrences\ of\ T_i\ in\ D}{Num\ of\ Occurrences\ of\ all\ Tokens\ in\ D}$$

We smooth the probabilities with Laplace smoothing [16] as follows:

$$P(T_i|C) = \frac{Num\ of\ Occurrences\ of\ T_i\ in\ D\ +\ 1}{Num\ of\ Occurrences\ of\ all\ Tokens\ in\ D\ +\ Num\ of\ Possible\ Token}$$



(a) (b)

Figure 1. CDF for: (a) number of reviews per contributor, and (b) average review size (number of words) per contributor.

2.2 Kullback-Leibler Divergence Metric

Kullback-Leibler Divergence (KLD) metric measures the distance between two distributions. For any two distributions P and Q, it is defined as:

$$D_{kl}(P\|Q) = \sum_i P(i) log(\frac{P(i)}{Q(i)})$$

KLD is always positive: the closer to zero, the closer Q is to P. It is an asymmetrical metric, i.e., $D_{kl}(P\|Q) \neq D_{kl}(Q\|P)$. To transform it into a symmetrical metric, we use the following formula (that has been used in [23]):

$$SymD_{kl}(P, Q) = 0.5 \times (D_{kl}(P\|Q) + D_{kl}(Q\|P)) \quad (2)$$

Basically, $SymD_{kl}$ is a symmetrical version of D_{kl} that measures the distance between two distributions. As discussed below, it is used heavily in our study. In the rest of the paper, the term "KLD" stands for $SymD_{kl}$[1].

3. Data Set

Clearly, a sizable set of reviews authored by a large number of contributors is necessary in order to perform a meaningful study. For that, we collected $1,076,850$ reviews for $1,997$ contributors from `yelp.com`, a very popular site with many prolific contributors. As shown in Figure 1(a), the minimum number of reviews per contributor is 330, the maximum – $3,387$ and the average – 539 reviews, with a standard deviation of 354. For the purpose of this study, we limited authorship to prolific contributors, since this provides more useful information for the purpose of review linkage.

Figure 1(a) shows the Cumulative Distribution Function (CDF) of the number of reviews per contributor. 50% of the contributors authored fewer than 500 reviews and 76% authored fewer than 600. Only 6% of the contributors exceed $1,000$ reviews.

Figure 1(b) shows the CDF for average review size (number of words) per contributor. It shows that 50% of the

[1] Note that, under certain conditions, NB and asymmetrical KLD models could be equivalent. That is, $argmax_{Class} P(Class|T_1, T_2, ..., T_n)$ is equivalent to $argmin_{Class} D_{kl}(Token\_distribution\|Class\_distribution)$, where $T_1, T_2, ...T_n$ are the tokens of a document D and $Token\_distribution$ is their derived distribution. The proof for this equivalency is in [23]. However, this equivalence does not hold when we use the symmetrical version $SymD_{kl}$.

contributors write reviews shorter than 140 words (on average) and 75% – have average review size smaller than 185. Also, 97% of the contributors write reviews shorter than 300 words. The overall average review size is relatively small at 149 words.

4. Study Settings

As mentioned earlier, our central goal is to study linkability of prolific reviewers. Specifically, we want to understand – for a given prolific author – to what extent some of his/her reviews relate to, or resemble, others. To achieve that, we first randomly sort the reviews of each contributor. Then, for each contributor U with N_U reviews, we split the randomly sorted reviews into two sets:

1. First $N_U - X$ reviews: We refer to it as the **identified record** of U.
2. Last X reviews: These reviews represent the full set of anonymous reviews of U from which we derive several subsets of various sizes. We refer to each of these subset as an **anonymous record** of U. An **anonymous record** of size i consists of the first i reviews of the full set of anonymous reviews of U. We vary the size of the **anonymous records** for the purpose of studying the user reviews linkability under different number of anonymous reviews.

Since we want to restrict the size of the anonymous set of reviews to a small portion of the complete user reviews set, we restrict X to 60 as this represents less than 20% of the minimum number of reviews for authors in our set (330). We use the **identified records** of all contributors as the training set upon which we build models for linking anonymous reviews. Note that the size of **identified record** is not the same for all contributors, while the sizes of **anonymous records** of every user are uniform.

Thus, our problem is reduced to matching an anonymous record to its corresponding identified record. Specifically, one anonymous record serves as input to a matching/linking model and the output is a sorted list of all possible account-ids listed in descending order of probability, i.e., the top-ranked account-id corresponds to the contributor whose identified record represents the most probable match for the input anonymous record. Then, if the correct account-id of the actual author is among top T entries, the matching/linking model has a hit; otherwise, it is a miss. Consequently, our study boils down to exploring matching/linking models that maximize the hit ratio of the anonymous records for varying values of both T and anonymous record sizes. We consider three values of T: 1 (perfect hit), 10 (near-hit) and 50 (near-miss). Whereas, for the anonymous record size, we experiment with a wider range of values: 1, 5, 10, 20, 30, 40, 50 and 60.

Even though the concentration of the paper is the study of the linkability of prolific users, we attempt to examine the performance of our models in non-prolific case. Thus, we slightly change the problem settings where we restrict the size of the **identified record** to smaller sizes. For clarity, we defer the description to Section 5.3.4 where we examine the non-prolific case.

NB	Naïve Bayes Model
KLD	Symmetrical Kullback-Leibler Divergence Model
R	Token Type: rating, unigram or digram
LR	Linkability Ratio
AR	Anonymous Record
IR	Identified Record (corresponding to a certain reviewer)
$SymD_{KLD}(IR, AR)$	symmetric KLD distance between IR and AR
$SymD_{KLD\_r}$	symmetric KLD of rating tokens
$SymD_{KLD\_c}$	symmetric KLD of category tokens
$SymD_{KLD\_l}$	symmetric KLD of lexical(unigram or digram) tokens
$SymD_{KLD\_r\_c}$	symmetric KLD of rating and category tokens
$SymD_{KLD\_l\_r\_c}$	symmetric KLD of lexical, rating and category tokens

Table 1. Notation and abbreviations.

5. Analysis

As mentioned in Section 2, we use Naïve Bayes (NB) and Kullback-Leibler Divergence (KLD) models. Before analyzing the collected data, we tokenize all reviews and extract four types of tokens:

1. **Unigrams:** set of all single letters. We discard all non-alphabetical characters.
2. **Digrams:** set of all consecutive letter-pairs. We discard all non-alphabetical characters.
3. **Rating:** rating associated with the review. (In Yelp, this ranges between 1 and 5).
4. **Category:** category associated with the place/service being reviewed. There are 31 categories in our dataset,

Why such simple tokens? Our choice of these four primitive token types might seem trivial or even naïve. In fact, initial goals of this study included more "sophisticated" types of tokens, such as: (1) distribution of word usage, (2) sentence length in words, and (3) punctuation usage. We originally planned to use unigrams and digrams as a baseline, imagining that (as long as all reviews are written in the same language – English, in our case) single and double-letter distributions would remain more-or-less constant across contributors. However, as our results clearly indicate, our hypothesis was wrong.

In the rest of this section, we analyze results produced by NB and KLD models. Before proceeding, we re-cap abbreviations and notation in Table 1.

5.1 Methodology

We begin with the brief description of the methodology for the two models.

5.1.1 Naïve Bayes (NB) Model

For each account IR, we built an NB model, $P(token_i|IR)$, from its identified record. Probabilities are computed using the Maximum-Likelihood estimator [5] and Laplace smoothing [16] as shown in 2. We then construct four models corresponding to four aforementioned token types. That is, for each IR, we have $P_{unigram}$, P_{digram}, $P_{category}$ and P_{rating}.

To link an anonymous record AR to an account IR with respect to token type R, we first extract all R-type tokens from AR, $T_{R_1}, T_{R_2}, ...T_{R_n}$ (Where T_{R_i} is the i-th R token in AR). Then, for each IR, we compute the probability $P_R(IR|T_{R_1}, T_{R_2}, ...T_{R_n})$. Finally, we return a list of accounts sorted in decreasing order of probabilities. The top entry represents the most probable match.

5.1.2 Kullback-Leibler Divergence (KLD) Model

We use symmetric KLD (see Section 2) to compute the distance between anonymous and identified records. To do so, we first compute distributions of all records, as follows:

$$Dist\_token(Token_i) = \frac{Num\ of\ Occurrences\ of\ Token_i}{Num\ of\ Occurrences\ of\ all\ Tokens}$$

To avoid division by 0, we smooth distributions via Laplace smoothing [16], as follows:

$$Dist\_token(Token_i) = \frac{Num\ of\ Occurrences\ of\ Token_i\ +\ 1}{Num\ of\ Occurrences\ of\ all\ Tokens\ +\ Num\ of\ Possible\ Token\ Values}$$

As before, we compute four distributions. To link AR with respect to token type R, we compute $SymD_{kl}$ between the distribution of R for AR and the distribution of R for each IR. Then, we return a list sorted in ascending order of $SymD_{KLD}(IR, AR)$ values. The first entry represents the account with the most likely match.

5.2 Study Results

We now present the results corresponding to our four tokens. Then, in the next section, we experiment with some combinations thereof.

5.2.1 Non-Lexical: Rating and Category

Figure 2 shows Top-1, Top-10, and Top-50 plots of the linkability ratios (LRs) for NB and KLD models for several anonymous record sizes when either rating or category is used as the token. Not surprisingly, an increase in the anonymous record size causes an increase in LR. Figures 2(a) and 2(b) show LR when rating token alone is used. In the Top-1 plot, LR is low and the highest ratio is 2.5%/1.9% in NB/KLD for anonymous record size of 60. However, in Top-10 and Top-50 plots, LRs become higher and reach 13.9%(11.9%) and 34.8%(33.3%) in Top-10 and Top-50 plots, respectively, in NB(KLD) for the same anonymous record size. Figures 2(c) and 2(d) show LR for the category token. In Top-1, the highest LR is 13.6%/4.7% in NB/KLD for anonymous record size of 60. A significant increase occurs in LRs in Top-10 and Top-50 plots: 40.4%(17.4%) and

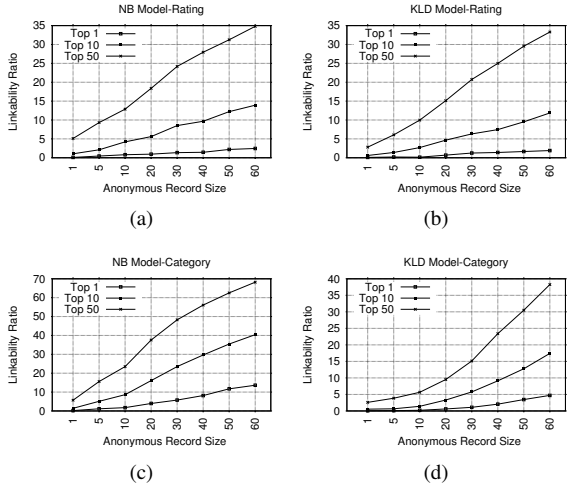


Figure 2. LRs for NB and KLD models rating and category tokens

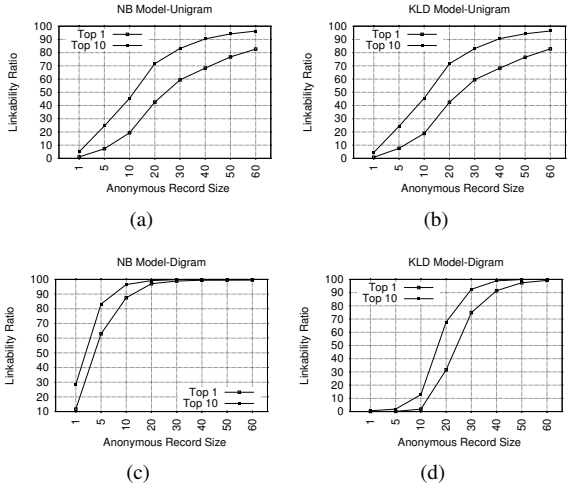


Figure 3. LRs of NB and KLD models for unigrams and digrams

68.2%(38.3%), respectively, in NB(KLD) model. The category is clearly more effective than the rating token. Additionally, we observe that NB performs better than KLD model, especially, for the category token.

We conclude that rating- and category-based models are only somewhat helpful, yet insufficient to link accounts for many anonymous records. However, it turns out that they are quite useful when combined with other lexical tokens, as discussed in Section 5.3 below.

5.2.2 Lexical: Unigram and Digram

Figure 5.2.2 shows the results for lexical tokens. Figures 3(a) and 3(b) depict LRs (Top-1 and Top-10) for NB and KLD with the unigram token. As expected, with the increase in anonymous record size, LR grows: it is high in both Top-

1 and Top-10 plots. For example, in Top-1 of both figures, LRs are around: 19%, 59% and 83% for anonymous record sizes of 10, 30 and 60, respectively. Whereas, in Top-10 of both figures, LRs are around: 45.5%, 83% and 96% for same record sizes. This suggests that reviews are highly linkable based on trivial single-letter distributions. Note that two models exhibit similar performance.

Figures 3(c) and 3(d) consider the digram token. In both models, LR is impressively high: it gets as high as 99.6%/99.2% in Top-1 for NB/KLD for AR size of 60. For example, Top-1 LRs in NB are: 11.7%, 62.9%, 87.5% and 97.1%, for respective AR sizes of 1, 5, 10 and 20. Whereas, in KLD, Top-1 LRs for record sizes of 10, 30 and 60 are: 1.9%,74.9% and 99.2%, respecitvely.

Unlike unigrams – where LRs in both models are comparable – KLD in digram starts with LRs considerably lower than those of NB. However, the situation changes when record size reaches 50, with KLD performing comparable to NB. One reason for that could be that KLD improves when the distribution of ARs is more similar to that of corresponding identified records; this usually occurs for large record sizes, as there are more tokens.

Not surprisingly, in both lexical and non-lexical models, larger AR sizes entail higher LRs. With NB, larger record size implies that, a given AR has more tokens in common with the corresponding IR. Thus, an increase in prediction probability $P(IR|T_1, T_2, ...T_n)$. For KLD, larger record size causes the distribution derived from AR to be more similar to the one derived from the corresponding IR.

5.3 Improvement I: Combining Lexical with non-lexical Tokens

In an attempt to improve LRs, we now combine the non-lexical token with is lexical counterparts.

5.3.1 Combining Tokens Methodology

This is straightforward in the NB. We simply increase the list of tokens in unigram- or digram-based NB by adding the non-lexical tokens. Thus, for every AR, we have $P(lexical\_token_i|IR)$, $P(category\_token_i|IR)$ and $P(rate\_token_i|IR)$.

Combining non-lexical with lexical tokens in KLD is less clear. One way is to simply average $SymD_{KLD}$ values for both token types. However, this might degrade performance, since lexical distributions convey much more information than their non-lexical counterparts. Thus, giving them the same weight would not yield better results. Instead, we combine them using a weighted average. First, we compute the weighted average of rating and category $SymD_{KLD}$:

$$SymD_{KLD\_r\_c}(P,Q) = \beta \times SymD_{KLD\_r}(P,Q) + (1-\beta) \times SymD_{KLD\_c}(P,Q)$$

Then, we combine the above with $SymD_{KLD}$ of the lexical tokens to compute the final weighted average:

$$SymD_{KLD\_l\_r\_c}(P,Q) = \alpha \times SymD_{KLD\_l}(P,Q) + (1-\alpha) \times SymD_{KLD\_r\_c}(P,Q)$$

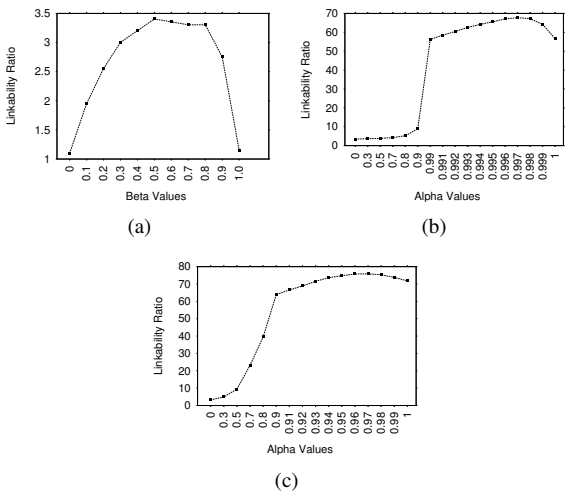


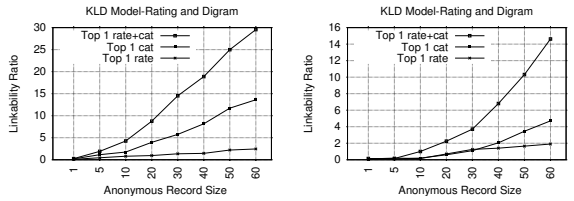
Figure 4. Results of combining different tokens using different β and α values

Thus, our goal is to get the right β and α values. Intuitively, $SymD_{kl\_lexical}$ should have more weight as it carries more information. Since there is no clear way of assigning weight values, we experimented with several choices and picked the one with the best performance; we discuss the selection process below. We experimented only within the IR set and then verified the results generalize to the AR. This was done as follows:

First, for every IR, we allocated the last 30 reviews as a testing record and the remainder – as a training record. Then, we expereminted with $SymD_{KLD\_r\_c}$ using several β values and set β to the value that yielded the highest LR based on the tested records. Then, we experimented with $SymD_{KLD\_l\_r\_c}$ using several α values and, similarly, picked the one with the highest LR.

Since β or α could assume any values, we need to restrict their choices. For β, we postulate that its optimal value is close to 0.5 since LRs for rating and category are comparable. Thus, we experimented with a range of values, from 0 to 1.00 in 0.1 increments. For α, we expect the optimal value to exceed 0.9, since LR for lexical tokens is significantly higher than for non-lexical ones. Therefore, we experimented with the weighted average by varying α between 0.9 and 0.99 in 0.01 increments.

If the values exhibit an increasing trend (i.e., $SymD_{KLD\_l\_r\_c}$ at α of 0.99 is the largest in this range) we continue experimenting in the $0.99 - -1.00$ range in 0.001 increments. Otherwise, we stop. For further verification, we also experimented with smaller α values: $0.0, 0.3, 0.5, 0.7$, and 0.8, all of which yielded LRs significantly lower than 0.9 for both unigram and digram. We acknowledge that we may be missing α or β values that could further optimize $SymD_{KLD\_l\_r\_c}$. However, results in Section **??**, show that our selection yields good results.



(a) (b)

Figure 5. LRs of NB and KLD for combining ratings and categories

Figure 4(a) shows LR (Top-1) for β values. The LR gradually increases until it tops off at 3.4% with $\beta = 0.5$ and then it gradually decreases. Figure 4(b) shows LR (Top-1) for α values in the unigram case. The LR has an increasing trend until it reaches 67.8% with $\alpha = 0.997$ and then it decreases. Figure 4(c) shows LR (Top-1) for α values in the diagram case where it tops off at 75.9% with $\alpha = 0.97$. Thus, the final values are 0.5 for β and $0.997/0.97$ for $alpha$ in unigram/digram case. Even though we extract α and β values by testing on a record size of 30, the results in following sections show that the derived weights are effective when tested on AR of other sizes.

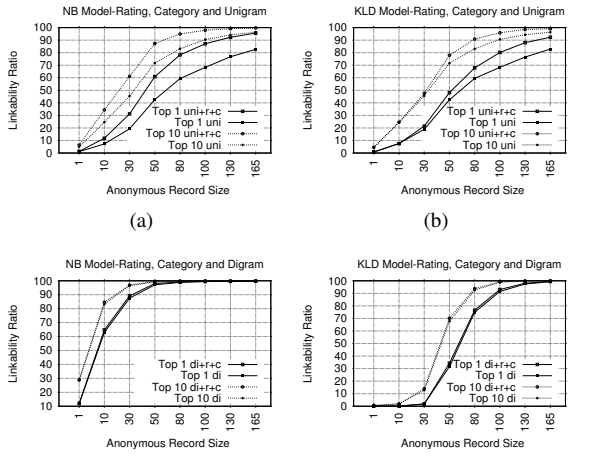
5.3.2 Combining Rating and Category - Results

Figures 5(a) and 5(b) show Top-1 plots for NB and KLD models when rating and category tokens are combined or used separately. Clearly, combining the tokens significantly improves LR in several record sizes. In NB, the gain in Top-1 LR ranges from 2.5-15.9%/3.5-27.1% over the category/rating based model for most record sizes. For example, LR increases from 5.8(1.4)% and 13.6(2.5)% in category(rating) based model to 29.5% and 14.5% in NB combined model for record sizes of 60 and 30, respectively. In Top-50, LR could reach as high as 87.7%, versus 68.2(34.8)% in the category (rating) based model for record size of 60.

In KLD, the gain in Top-1 LR ranges from 1.7-9.9%/1.6-12.7% over category/rating based model for most record sizes. For example, it leaps from 1.1(1.3)% and 4.7(1.9)% in category (rating) based model to 3.7% and 14.6% in KL combined model for record sizes of 60 and 30, respectively. The gain is even higher in Top-50 where it reaches 69.1%, versus 38.3(33.3)% in the category (rating) based model for record size of 60. These results shows that combining rating and category tokens is very effective in increasing LR in both NB and KLD models.

5.3.3 Combining Lexical with Non-Lexical Tokens

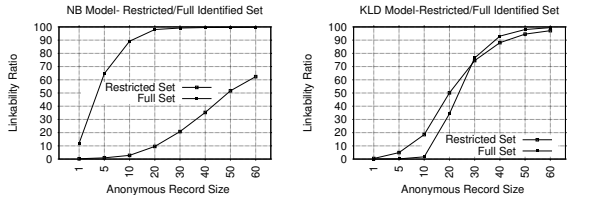
Figures 6(a) and 6(b) show Top-1 and Top-10 plots in NB and KLD models of unigram tokens before and after combining them with rating and category tokens. Adding non-lexical tokens to unigrams substantially increases LR in several record sizes. In NB, the gain in Top-1 LR ranges from



(a) (b)

(c) (d)

Figure 6. LRs for NB and KLD for combining ratings and categories with unigrams or digrams



(a) (b)

Figure 7. LRs for NB and KLD in full and restricted identified set

4.5-18.9% (1.4 - 15.7% for Top-10 LR). In KLD, the gain in Top-1 LR ranges from 2.5-11.9 (2-7.8% in Top-10 LR). These findings shows how effective is combining the non-lexical tokens with the unigrams. In fact, we can accurately identify almost all ARs.

Figures 6(c) and 6(d) show the effect of adding ratings and categories to digrams. The overall effect is minuscule: in NB (KLD) model, the increase in Top-1 LR ranges from 0.3-1.8% (0.2-2.7%) for most record sizes. The increase is very similar in Top-10 plots.

5.3.4 Restricting Identified Record Size

In previous sections, our analysis was based on using the full data set. That is, except for the anonymous part of the data set, we use all of user reviews as part of our identified set. Although LR is high in many cases, it is not clear how the models will perform when we restrict IR size. To this end, we re-evaluate the models with the same problem settings, however, with a restricted IR size. We restrict IR size to AR size; both randomly selected without replacement.

Figures 7(a) and 7(b) show two Top-1 plots in NB and KLD models: one plot corresponds to the restricted identified set and the other – to the full set. Tokens used in the

models consist of digrams, ratings and categories (since this combination gives the highest LR). Unlike the previous sections, where NB and KLD behaved similarly, the two models now behave differently when restricting the identified set. While NB performs better KLD on the full set, the latter performs much better than NB when the identified set is restricted. In fact, in some cases, KLD performs better when the set is restricted.

The reason for this improved KLD performance might be the following: in symmetric KLD distance function, the distributions of both IR and AR have to be very close in order to match regardless of the size of the IR; unlike the NB, where larger training sets would lead to better estimate of the token probabilities and thus more accurate predictions.

In KLD, we achieve high LR for many record sizes. For example, Top-1 LR in the restricted set is 74.5%, 88% and 97.1% when the anonymous (and identified) record sizes are 30, 40 and 60, respectively. Whereas, LR in the full set for the same AR sizes is: 76.5% , 93% and 99.4%. When the record size is less than 30, KLD performs better in the restricted set than the full one. For example, when AR size is 20, LR in the restricted set is 50.1% and 34.3% in the full set. In NB, Top-1 LR in the restricted set is lower than the full set. For instance, it is 20.8%, 35.3% and 62.4% for AR size of: 30, 40 and 60, respectively. Whereas, for the same sizes, LR is more than 99% in the full set.

This result has one very important implication: even with very small IR sizes, many anonymous users can be identified. For example, with only IR and AR sizes of only 30, most users can be accurately linked (75% in Top-1 and 90% in Top-10). This situation is very common since many real-world users generate 30 or more reviews over multiple sites. Therefore, even reviews from non-prolific accounts can be accurately linked.

5.3.5 Improvement II: Matching all ARs at Once

We now experiment with another natural strategy of attempting to match all ARs at once.

5.3.6 Methodology

In the previous section, we focused on linking one AR at a time. That is, ARs were independently and incrementally linked to IRs (accounts/reviewer-ids). One natural direction for potential improvements is to attempt to link all ARs at the same time. To this end, we construct algorithm $Match\_All()$ in Figure 8 as an add-on to the KLD models suggested in previous sections.

$SymD_{KLD}(IR_j, AR_i)$ symmetrically measures the distance between their (IR_j's and AR_i's) distributions. Since every AR maps to a distinct IR (AR_i maps to IR_i), it would seem that lower $SymD_{KLD}$ would lead to better match. We use this intuition to design $Match\_All()$. As shown in the figure, $Match\_All()$ picks the smallest $SymD_{KLD}(IR_j, AR_i)$ as the map between IR_j and AR_i and then deletes the pair (IR_j, V_{kj}) from all remaining lists in S_L. The process continues until we compute all matches. Note that, for any $List_{AR_k}$, (IR_j, V_{kj}) is deleted from the list only when there is another pair (IR_j, V_{lj}) in $List_{AR_l}$, such that $SymD_{KLD}(IR_j, AR_l) \leq SymD_{KLD}(IR_j, AR_k)$, and IR_j has been selected as the match for AR_l.The output of the algorithm is a match-list: $S_M = \{(IR_{i_1}, AR_{j_1}), (IR_{i_n}, AR_{j_n})\}$.

Algorithm $Match\_All$**:** Pseudo Code

Input: (1) Set of ARs: $S_{AR} = \{AR_1, AR_2, ..., AR_n\}$
(2) Set of reviewer-ids / identified records: $S_{IR} = \{IR_1, IR_2, ..., IR_n\}$
(3) Set of matching lists for each AR: $S_L = \{List_{AR_1}, .., List_{AR_n}\}$
Output: Matching list: $S_M = \{(IR_{i_1}, AR_{j_1}), (IR_{i_n}, AR_{j_n})\}$
set $S_M = \emptyset$
While $|S_{AR}| \neq 0$:
    Find AR_i with smallest $SymD_{KLD}$ value in all lists in S_L
    Get corresponding reviewer-id IR_j
    Add (IR_j, AR_i) to S_M
    Delete AR_i from S_{AR}
    Delete $List_{AR_i}$ from S_L
    For each $List_t$ in S_L,
        Delete tuple containing IR_j from $List_t$
    End For
End While

NOTE 1: $List_{AR_i}$ in S_L is a list of pairs (IR_j, V_{ij}) where $V_{ij} = SymD_{KLD}(IR_j, AR_i)$
NOTE 2: $List_{AR_i}$ is sorted in increasing order of V_{ij}, i.e., IR_j with lowest $SymD_{KLD}(IR_j, AR_i)$ at the top.

Figure 8. Pseudo-Code for matching all ARs at once.

We now consider how $Match\_All()$ could improve LR. Suppose that we have two ARs: AR_i and AR_j along with corresponding sorted lists L_i and L_j and assume that IR_i is at the top of each list. Using only KLD, we would return IR_i for both ARs and thus miss one of the two. Whereas, $Match\_All$, would assign IR_i to **only** one AR – the one with the smaller $SymD_{KLD}(IR_i, ...)$ value. We would intuitively suspect that $SymD_{KLD}(IR_i, AR_i) < SymD_{KLD}(IR_i, AR_j)$ since IR_i is the right match for AR_i and thus their distributions would probably be very close. If this is the case, $Match\_All$ would delete IR_i (erroneous match) from the top of L_j which could help clearing up the way for IR_j (correct match) to the top of L_j.

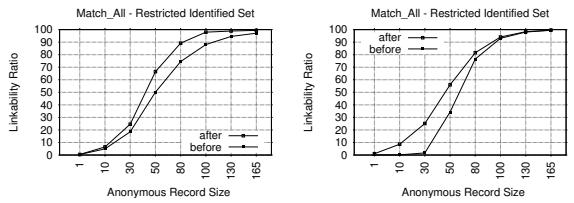
We note that there is no guarantee that $Match\_All()$ will always work: one mistake in early rounds would lead to others in later rounds. We believe that $Match\_All()$ works better if $SymD_{KLD}(IR_i, AR_i) < SymD_{KLD}(IR_j, AR_i)$ ($j \neq i$) holds most of the time.

In the next section, we show the results of $Match\_All()$ when we experiment with the KLD model with digram, rating and category tokens.[2].

5.3.7 Results

Figures 9(a) and 9(b) show the effect of $Match\_All()$ on Top-1 LR in both restricted identified set and full identi-

[2] We also tried $Match\_All()$ with the NB model and it did not improve LR.



(a) (b)

Figure 9. Effects of $Match\_All()$ on LRs in full and restricted identified set: before and after plots

fied set, respectively. The combination of diagram, rating and category tokens are used. Each figure shows two Top-1 plots: one for LR after using $Match\_All$ and the other – for LR before using it. Clearly, $Match\_All$ is effective in improving LR for almost all record sizes. For the restricted set, the gain in LR ranges from 1.6-16.4% for nearly all AR sizes. Similar increase is observed in the full set that ranges from 1-23.4%. This shows that the $Match\_All$ is very effective when used with diagram, rating and category tokens. The privacy implication of $Match\_All$ is important as it significantly increases LR for small ARs in the restricted set. This shows that privacy of non-prolific users is exposed even more with $Match_all$.

5.4 Study Summary

We now summarize the main findings and conclusions of our study.

1. LR becomes very high – reaching up to $\sim$ 99.5% in both KLD and NB when using only digram tokens. (See Section 5.2.2).
2. Surprisingly, using only unigramss, we can link up to 83% in both NB and KLD models, with 96% in Top-10. (See Section 5.2.2). This suggests that reviewers expose a great deal merely from their single letter distributions.
3. Even with small record sizes, we accurately link a significant ratio of ARs. Specifically, for AR size of 5 and 10 (using NB with diagrams), we can accurately link 63% and 88% ARs, respectively. (See Section 5.2.2).
4. Rating and category tokens are more useful if combined where 88%/69% of AR (size 60) fall into Top-50 in NB/KLD. (See Section 5.3.2).
5. Non-lexical tokens are very useful in tandem with lexical tokens, especially, the unigram: we observe a $\sim$19%/12% Top-1 LR increase in NB/KLD for some cases. (See Section 5.3.3).
6. Relying only on unigram, rating and category tokens, we can accurately link 96%/92% AR (size 60) in NB/KLD. (See Section 5.3.3).
7. Restricting IR size does not always degrade linkability. In KLD, we can link as many as 96% ARs when IR size is small. (See Section 5.3.4).
8. Linking all ARs at once (instead of each independently) helps improve accuracy. The gain is up to 15%. (See Section 5.3.7).
9. Generally, NB performs better than KLD when we use the full identified set and KLD performs better when we use the restricted identified set.

6. Discussion

Implications: We believe that the results of, and techniques used in, this study have several implications. First, we demonstrated the practicality of cross-referencing accounts (and reviews) among multiple review sites. If a person contributes to two sites under two identities, it is highly likely that sets of reviews from these sites can be linked. This could be quite detrimental to contributors' privacy.

The second implication is the ability to correlate – on the same review site – multiple accounts that are in fact manipulated by the same person. This could make our techniques very useful in detecting review spam [11], whereby a contributor authors reviews under different accounts to tout (also self-promote) or criticize a product or a service.

Prolific Users: While there are clearly many more occasional (non-prolific) reviewers than prolific ones, we believe that a study on prolific reviewers is important, for two reasons. First, the number of prolific contributors is quite large. For example, from only one review site, we identified $\sim$ 2,000 such reviewers. Second, given the spike of popularity of review sites [2], we believe that, in the near future, the number of contributors will grow substantially. and that will lead to an increasing number of productive reviewers. Also, even many occasional reviewers, with the passage of time, will enter the ranks of "prolific" ones, i.e., by slowly accumulating a sufficient corpus of reviews over the years. Nevertheless, our study suggests that even non-prolific users have their privacy compromised(see Section 5.3.5). For example, when both the identified and anonymous records are of size 20 reviews (total user contribution is 40 reviews), we are able to accurately link half of the anonymous records to their reviewers. Additionally, the reviewers of most the anonymous records are listed among the Top-10 output of the models.

Anonymous Record Size: Our models perform best when the AR size is of 60 reviews. However, for every reviewer in our dataset, 60 is less than 20% of that person's total number of reviews;i.e., a small portion of users contribution. Additionally, using the NB model along with digram, rating and category tokens, we can accurately link most of the anonymous records when their size is only 10. An anonymous record of size 10 represents only 3% of the minimum user contribution.

Unigram Tokens: While our best performing models are based on the digram tokens, we are able to get high linkability results from unigram tokens that reach up to 83% (96% in the Top 10) in NB or KLD model. The results are improved to 96/92% in NB/KLD model when we combine the unigram tokens with rating and category tokens. Note that the number of tokens in the unigram based models is 59/26 tokens with/without combining the unigrams with rating and category tokens while the number of tokens in the diagram based models is 676 tokens and 709 tokens when combined with rating and category tokens. This makes the accuracy of the attack (to link anonymous reviews) that is based on unigram models very comparable to its diagram counterpart while the number of tokens is significantly less. This implies a substantial reduction in the resources and processing power in the unigram-based models which would make them scale better. For example, assume the attacker takes over a set of anonymous reviews and wants to link them to many large review datasets instead of one. In this scenario, the unigram based models would scale better while maintaining the same level of accuracy.

Counter Measures: One concrete application of our techniques is via integration with review site's front-end software in order to provide feedback to authors indicating the degree of linkability of their reviews. For example, when the reviewer log into the site, a linkability nominal/categorical value (e.g. high, medium, and low) is shown to the user indicating how his/her last 5 or 10 reviews are linkable to his/her old ones. It would then be up to to the individual to maintain or modify their reviewing patterns (or delete their reviews, if necessary) to be less linkable. Another way of countering against the linkability attacks is to make the reviewer systems automatically suggest a different choice of words to the users that are less revealing(less personal) and more common among different users. We suspect with the use of such words, reviews would be less linkable and the lexical distributions for different users would be more similar. Additionally, the review sites could provide the users an alternative way of describing their opinion without resorting to writing reviews. For example, the review sites may provide the user with a set of questions and a fixed set of answers and the user express his/her experience by choosing the suitable set of answers. We suspect that this would lead to less linkable reviews as the textual reviews might be more personal. We leave it to our future work to examine the usability and effectiveness of these techniques.

7. Future Work

Although our results point at high linkability of reviews among contributors, there remain many open questions. First, the anonymous records are not highly linkable when their sizes are restricted to 1. As part of our future work, we plan to improve the linkability on very small anonymous records. In addition, although we leverage ratings and categories to boost LRs, we need to further explore the effect of using other non-textual features such as the sub-categories of places, products and services reviewed and the length of the review other non-textual features such as the sub-categories places, products and services reviewed and the length of the review. In fact, it would be interesting to see how LR can be improved without resorting to lexical features (since lexical features generally entail heavy processing and large sets of features). We also plan to implement the counter measure techniques explained in Section 6 and examine their effectiveness and useabilities.

Moreover, we need to investigate LRs in other preference databases (such as music/song ratings) and check whether contributors inadvertently link their reviews through preferences. It would be interesting to see how to leverage techniques used in recommender systems(for future rating prediction) to increase LRs.

In Section 5.3.5, we show how to improve LRs by linking all the anonymous records at once. As part of our future work, we plan to further investigate the effect (on LR) of the number anonymous records when each record belongs to a different reviewer.

8. Related Work

Many authorship analysis studies have appeared in the literature. Among the most prominent recent studies are: [4, 10, 24]. The study in [10] proposes techniques that are relied on extracting frequent pattern write-prints that characterized one (or a group of) authors. The best achieved accuracy was 88% when identifying an author, from a single anonymous message, from a small set of four and with training set size of forty messages per author. In [24], a framework for author identification for on-line messages was introduced where four types of features were extracted: lexical, syntactic, structural and content-specific. Three types of classifiers were used for author identification: Decision Trees, Back Propagation Neural Networks and Support Vector machines. The last one outperformed the others achieving 97% in a set of authors that did not exceed 20. The work in [4] also considered author identification and similarity detection by incorporating a rich set of stylistic features along with a novel technique(based on Karhunen-Loeve-transforms) to extract write-prints. The techniques were shown to perform well and reached as high as 91% in identifying the author of anonymous text from a set of 100. The same approach was tested on a large set of Buyer/Seller Ebay feedback comments collected from Ebay. Such comments typically reflect one's experience when dealing with a buyer or a seller. Unlike our general-purpose reviews, these comments do not review products, services or places of different categories. Additionally, the scale of the problem was different and the analysis was performed for 100 authors, whereas, our analysis involved $\sim 2,000$ reviewers.

A problem very similar to ours was explored in [17]. It focused on identifying authors based on reviews in both single- and double-blinded peer-reviewing processes. Naïve Bayes classifier was used – along with unigrams, bigrams and trigrams – to identify authors and the best result was around 90%. In [8], citations of a given paper were used to identify its authors. The data set was a very large archive of physics research papers (KDDCUP 2003 physics-paper archive). Authors were identified 40-45% of the times. In [21], authorship analysis was performed on a set of candidate authors who wrote on the same topics. Specifically, analysis was done on movie reviews of five reviewers on the same five movies. Although reviews similar to ours were used, there were significant differences. We use over $1,000,000$ reviews by $\sim$ $2,000$ authors, whereas, only 25 reviews by 5 authors were used in [21]. A related result [19] studied the problem of inferring the gender of a movie reviewer from his/her review. Using logistic regression [15] along with features derived from the writing style, content, and meta-data of the review, accuracy of up to 73.7% was achieved in determining correct gender. The goal of this study was clearly quite different from ours. For a comprehensive overview of authorship analysis studies, we refer to [22].

While all aforementioned results are somewhat similar to our present work, there are some notable differences. First, we perform authorship identification analysis in a context that has not been extensively explored – user reviews. User reviews are generally written differently from other types of writing, such as email and research papers. In a review, the author generally assesses something and thus the text conveys some evaluation and personal opinions. A review usually conveys information about personal taste, since most people tend to review things of interest to them. In addition, reviews contain other non-textual information, such as the ratings and categories of things being reviewed. These types of extra information provide added leverage; recall that, as discussed in previous sections, ratings are particularly helpful in increasing the overall linkability ratio. Second, our problem formulation is different. We study linkability of reviews (and user-ids of their authors) in the presence of a large number of prolific contributors where the number of anonymous reviews could be more than one (up to 60 reviews). Whereas, most prior work attempts to identify authors from a small set of authors, each with small sets of texts where the number of anonymous documents/messages is one.

Some work has been done in recovering authors based on their ratings, using external knowledge. In particular, [7] studied author linkability with two different databases; one public and the other – private. Several techniques were used to link authors in public forums (public) who state their opinions and rating about movies to reviewers who contribute to a sparse database (private) of movie ratings. A related result [18] considered anonymity in high-dimensional and sparse data sets of anonymized users. First, it presented a general definition and model of privacy breaches in such sets.Second, a statistical de-anonymization attack was presented that was resilient to perturbation. Third, this attack was used to de-anonymize the Netflix [1] data set. Note that the problem formulation of these two results differs from ours. They studied anonymity in the presence of an external source of public information. Whereas, our work does not rely on any external sources.

Last but not least, another related effort was made to assess the authenticity of reviews [11]. It explored the problem of identifying spam reviews. Results demonstrated that spam reviews were prevalent and a counter-measure based on logistic regression was proposed.

9. Conclusion

Large numbers of Internet users are becoming frequent visitors and contributors to various review sites. At the same time, they are concerned about their privacy. In this paper, we study linkability of reviews. Based on a large set of reviews, we show that a high percentage (99% in some cases) are linkable, even though we use very simple models and very simple features set. Our study suggests that users reliably expose their identities in reviews. This has certain important implications for cross-referencing accounts among different review sites and detecting people who write reviews under different identities. Additionally, techniques used in this study could be adopted by review sites to give contributors feedback about linkability of their reviews.

References

[1] Netflix. http://www.netflix.com.

[2] Yelp By The Numbers. http://officialblog.yelp.com/2010/12/2010-yelp-by-the-numbers.html.

[3] Yelp Elite Squad. http://www.yelp.com/faq#what_is_elite_squad.

[4] A. Abbasi and H. Chen. Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace. In *ACM Transactions on Information Systems*, 2008.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] K. Dave, S. Lawrence, and D. M. Pennock. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *international conference on World Wide Web*, 2003.

[7] D. Frankowski, D. Cosley, S. Sen, L. Terveen, and J. Riedl. You Are What You Say: Privacy Risks of Public Mentions. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[8] S. Hill and F. Provost. The Myth of the Double-Blind Review?: Author Identification Using Only Citations. In *SIGKDD Explorations Newsletter*, 2003.

[9] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[10] F. Iqbal, H. Binsalleeh, B. Fung, and M. Debbabi. A unified data mining solution for authorship analysis in anonymous textual communications. In *Information Sciences (INS): Special Issue on Data Mining for Information Security*, 2011.

[11] N. Jindal and B. Liu. Opinion Spam and Analysis. In *ACM International Conference on Web Search and Data Mining*, 2008.

[12] N. Jindal, B. Liu, and E.-P. Lim. Finding Unusual Review Patterns Using Unexpected Rules. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010.

[13] D. Lewis. Naive(bayes) at forty:the independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, 1998.

[14] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. Lauw. Detecting Product Review Spammers using Rating Behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010.

[15] S. Menard. Applied Logistic Regression Analysis. In *Sage University Press*, 2002.

[16] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[17] M. Nanavati, N. Taylor, W. Aiello, and A. Warfield. Herbert West – Deanonymizer. In *6th USENIX Workshop on Hot Topics in Security*, 2011.

[18] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *IEEE Symposium on Security and Privacy*, 2009.

[19] J. Otterbacher. Inferring Gender of Movie Reviewers: Exploiting Writing Style, Content and Metadata. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010.

[20] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Empirical Methods on Natural Language Processing Conference*, 2002.

[21] D. S. Ross Clement. Ngram and Bayesian Classification of Documents for Topic and Authorship. In *Literary and Linguistic Computing*, 2003.

[22] E. Stamatatos. A Survey of Modern Authorship Attribution Methods. In *Journal of the American Society for Information Science and Technology*, 2009.

[23] S. Yadav, A. K. Reddy, A. N. Reddy, and S. Ranjan. Detecting Algorithmically Generated Malicious Domain Names. In *Internet Measurement Conference*, 2010.

[24] R. Zheng, J. Li, H. Chen, and Z. Huang. A Framework for Authorship Identification of Online Messages: Writing Style Features and Classification Techniques. In *Journal of the American Society for Information Science and Technology*, 2006.