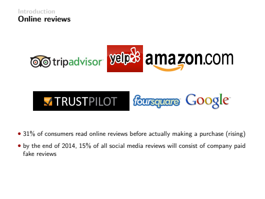
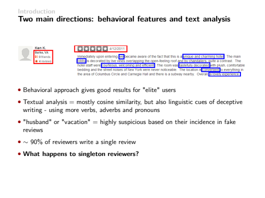


Introduction

Online reviews

- 31% of consumers read online reviews before actually making a purchase (rising)
- by the end of 2014, 15% of all social media reviews will consist of company paid fake reviews

Online reviews have become in recent years a very important resource for consumers when making purchases. The number of consumers that first read reviews about a product they wish to buy is constantly on the rise. Technology research company Gartner Inc. claims 31% of consumers read online reviews before actually making a purchase. As consumers increasingly rely on these ratings, the incentive for companies to try to produce fake reviews to boost sales is also increasing. Gartner predicts in 2014 as much as 15 percent of all social media reviews will consist of company paid fake reviews.



Introduction

Two main directions: behavioral features and text analysis

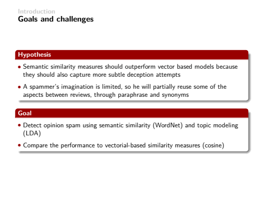
Ken K.
Burke, VA
0 friends
4 reviews

4/12/2011

Immediately upon entering we became aware of the fact that this is a unique and charming hotel. The main lobby is decorated by live vines overlapping the open-feeling roof and by chandaliers, quite a contrast. The hotel staff were courteous, welcoming and efficient. The room was tastefully decorated with plush, comfortable bedding and the street noises of New York were never noticeable. The location is convenient to everything in the area of Columbus Circle and Carnegie Hall and there is a subway nearby. Overall, a lovely experience.

- Behavioral approach gives good results for "elite" users
- Textual analysis = mostly cosine similarity, but also linguistic cues of deceptive writing - using more verbs, adverbs and pronouns
- "husband" or "vacation" = highly suspicious based on their incidence in fake reviews
- ∼ 90% of reviewers write a single review
- **What happens to singleton reviewers?**

The first method so far seems to be more reliable and can be more easily put into practice. It also offers very good results as a standalone method, although the textual features do bring little overall improvement. The linguistic techniques used so far mostly consisted of computing cosine similarity between the contents of the reviews. In a new study, the authors concluded that human judgment used to detect semantic similarity of web document does not correlate well with cosine similarity. Other researches used a bag-of-words approach and calculated the frequency of certain words from the review text. They then classified some reviews as suspicious if the text contained a high number of predefined suspicious words. This led to more subjective conclusions that spammers prefer to use more personal pronouns than genuine reviewers or they usually write reviews of more than 150 characters on average. An obvious aspect is that once the spammers find out about these textual frequency traps which cause suspicion, they will simply avoid them.



Introduction

Goals and challenges

Hypothesis

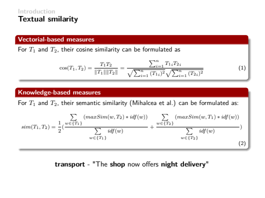
- Semantic similarity measures should outperform vector based models because they should also capture more subtle deception attempts
- A spammer's imagination is limited, so he will partially reuse some of the aspects between reviews, through paraphrase and synonyms

Goal

- Detect opinion spam using semantic similarity (WordNet) and topic modeling (LDA)
- Compare the performance to vectorial-based similarity measures (cosine)

This thesis proposes a complete solution to detect opinion spam of one-time reviewers using semantic similarity. It also proposes a method to detect opinion spam, using recent research models aimed at extracting product aspects from short texts, based on topic modeling and in particular on Latent Dirichlet Allocation (LDA). Another goal was to test this hypothesis on real-life reviews and make a comparison with the existing vectorial similarity models, which are based on cosine similarity.

My hypothesis is that semantic similarity measures should outperform vector based models because they should also capture more subtle deception behavior, meaning more paraphrase intent of the spammers. Detecting fake reviews through semantic similarity methods would inherently work on users who operate in groups, know one other and paraphrase or rephrase each other's reviews inside the same group of spammers.



Introduction

Textual smilarity

Vectorial-based measures

For T_1 and T_2, their cosine similarity can be formulated as

$$\cos(T_1, T_2) = \frac{T_1 T_2}{\|T_1\| \|T_2\|} = \frac{\sum_{i=1}^{n} {T_1}_i {T_2}_i}{\sqrt{\sum_{i=1}^{n} ({T_1}_i)^2} \sqrt{\sum_{i=1}^{n} ({T_2}_i)^2}} \tag{1}$$

Knowledge-based measures

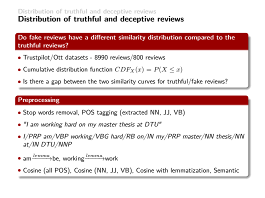
For T_1 and T_2, their semantic similarity (Mihalcea et al.) can be formulated as:

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{w \in \{T_1\}} (maxSim(w, T_2) * idf(w))}{\sum_{w \in \{T_1\}} idf(w)} + \frac{\sum_{w \in \{T_2\}} (maxSim(w, T_1) * idf(w))}{\sum_{w \in \{T_2\}} idf(w)}\right) \tag{2}$$

transport - "The **shop** now offers **night delivery**"

Given a metric for word-to-word similarity(WordNet) and a measure of word specificity(idf)(no. of documents in corpus / no. of documents containing the word), we define the semantic similarity of two text segments T1 and T2 using a metric that combines the semantic similarities of each text segment in turn with respect to the other text segment.
First, for each word w in the segment T1 we try to identify the word in the segment T2 that has the highest semantic similarity (maxSim(w,T2)). Next, the same process is applied to determine the most similar word in T1 starting with words in T2. The word similarities are then weighted is applied to determine the most similar word in T1 starting with words in T2. The word similarities are then weighted with the corresponding word specificity, summed up, and normalized with the length of each text segment. Finally the resulting similarity scores are combined using a simple average. Only nouns and verbs are matched, for adjectives and adverbs - lexical matching is used.

This means that, for instance, the most similar word to the noun "transport" within the text "The shop now offers night delivery" will be sought among the nouns "shop", "night" and "delivery", and will ignore the words with a different part-of-speech - "offers", "now".



Distribution of truthful and deceptive reviews

Distribution of truthful and deceptive reviews

Do fake reviews have a different similarity distribution compared to the truthful reviews?

- Trustpilot/Ott datasets - 8990 reviews/800 reviews
- Cumulative distribution function $CDF_X(x) = P(X \leq x)$
- Is there a gap between the two similarity curves for truthful/fake reviews?

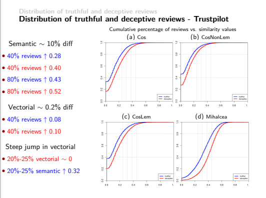
Preprocessing

- Stop words removal, POS tagging (extracted NN, JJ, VB)
- *"I am working hard on my master thesis at DTU"*
- *I/PRP am/VBP working/VBG hard/RB on/IN my/PRP master/NN thesis/NN at/IN DTU/NNP*
- am$\xrightarrow{lemma}$be, working$\xrightarrow{lemma}$work
- Cosine (all POS), Cosine (NN, JJ, VB), Cosine with lemmatization, Semantic

Is there a distributional difference, for both vectorial and semantic similarity measures, between deceptive and truthful reviews inside the Trustpilot dataset as well as in the well known dataset used by Ott. Recall that Ott obtained deceptive reviews using Amazon Mechanical Turk - the turkers had to imagine staying at the specific hotels and write their reviews to sound as credible as possible.

One research study claims that spammers caught by Yelp's filter seem to have "overdone faking" in their attempt to sound more genuine. In their deceptive reviews, they used words that appeared in genuine reviews almost equally frequent and avoided to reuse the exact same words over and over again in their deceptive reviews. This is exactly the reason why a cosine similarity measure is not enough to catch subtle spammers in real life scenarios, such as Yelp's.

The CDF gives the probability of a random variable X, which has a known probability distribution, to have a value less or equal to x. The purpose of the CDF curves of truthful/fake reviews is to check if they overlap or there is a separation margin between the two curves. If indeed there is a gap, it would be interesting to know how big this separation is and what are its bounds.



Distribution of truthful and deceptive reviews

Distribution of truthful and deceptive reviews - Trustpilot

Semantic ~ 10% diff

- 40% reviews ↑ 0.28
- 40% reviews ↑ 0.40
- 80% reviews ↑ 0.43
- 80% reviews ↑ 0.52

Vectorial ~ 0.2% diff

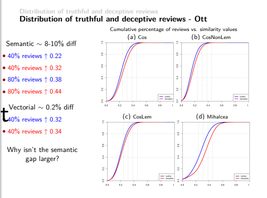
- 40% reviews ↑ 0.08
- 40% reviews ↑ 0.10

Steep jump in vectorial

- 20%-25% vectorial ~ 0
- 20%-25% semantic ↑ 0.32

Cumulative percentage of reviews vs. similarity values

The CDF curves plotted reveal several interesting aspects. They show the amount of content similarity for the truthful/fake reviews taken separately as well as the position and bounds for each type and the gaps between the two curves. Regardless of the type of similarity measure used, i.e. vectorial or semantic, the two distributional curves of truthful and fake reviews are clearly separated. For the truthful reviews (blue color), the curve appears towards the left of the plot, while for the fake reviews (red color) it is more towards the right. This means that for any similarity measure applied, for a fixed cumulative percentage value, its corresponding value x_t for truthful reviews will be lower than the value for deceptive reviews x_d. This shows that people writing deceptive reviews tend to have a higher semantic similarity between their reviews than the users writing honest reviews, probably because they are the same person under multiple accounts or know each other and work together. The honest users do not influence each other's writing style so much. The spammers are more likely to do that.



Distribution of truthful and deceptive reviews

Distribution of truthful and deceptive reviews - Ott

Semantic ∼ 8-10% diff

- 40% reviews ↑ 0.22
- 40% reviews ↑ 0.32
- 80% reviews ↑ 0.38
- 80% reviews ↑ 0.44

Vectorial ∼ 0.2% diff

- 40% reviews ↑ 0.32
- 40% reviews ↑ 0.34

Why isn't the semantic gap larger?

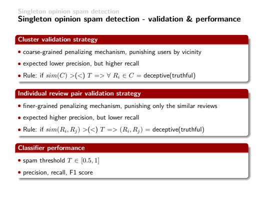
Cumulative percentage of reviews vs. similarity values

(a) Cos (b) CosNonLem (c) CosLem (d) Mihalcea

One obvious question is why isn't this gap larger, especially for the semantic similarity measure, regardless of the dataset?

One possible explanation might be that reviews from the same seller generally talk pretty much about aspects within a very specific context, which is related to the shop's business area of activity. For example, if the shop is an electronics reseller that offers online ordering, home delivery and customer support for sold items, then the review will probably contain aspects related to website, the delivery speed, customer support, service level, screen, battery, price and so on. It is pretty easy for the spammers to mimic the honest reviews in the sense of mentioning the same key aspects in their reviews.

The dataset of Ott has been created using Amazon Mechanical Turk (AMT), so it is likely the turkers was separate persons and did not know each other. It is a different setup than for the real-life Trustpilot dataset, where the fake reviews are more likely to be written by the same people using multiple accounts on the review platform.



Singleton opinion spam detection

Singleton opinion spam detection - validation & performance

Cluster validation strategy

- coarse-grained penalizing mechanism, punishing users by vicinity
- expected lower precision, but higher recall
- Rule: if $sim(C) >(<) T => \forall\ R_i \in C =$ deceptive(truthful)

Individual review pair validation strategy

- finer-grained penalizing mechanism, punishing only the similar reviews
- expected higher precision, but lower recall
- Rule: if $sim(R_i, R_j) >(<) T => (R_i, R_j) =$ deceptive(truthful)

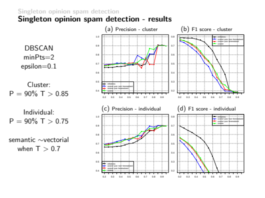
Classifier performance

- spam threshold $T \in [0.5, 1]$
- precision, recall, F1 score

The first strategy acts as a coarse-grained mechanism penalizing all the reviews in the cluster if any of the contained pairs scores a similarity above the threshold. So, users get penalized just by being in the close vicinity of highly suspicious users, by sharing the same cluster. Intuitively, this approach should give a lower precision than the individual review pair strategy, but a higher recall.

On the other hand, the individual review pair strategy should achieve the best precision because of its fine granularity. A more general mitigation approach could automatically filter out the review pairs scoring above the threshold and then consider the rest of their cluster neighbors as highly suspicious reviews, but apply more detection methods or manual validation before making a final decision to mark them as truthful or as deceptive.

Since I have tested multiple similarity methods between reviews, it is important to be able to say which one is actually the best at detecting fake reviews. And since the classifier performance is shown by two measures, I have used the F1 score to combine the two into a single value and then more easily choose the similarity method which performs best.



Singleton opinion spam detection

Singleton opinion spam detection - results

DBSCAN
minPts=2
epsilon=0.1

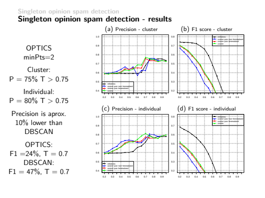
Cluster:
P = 90% T > 0.85

Individual:
P = 90% T > 0.75

semantic ∼vectorial
when T > 0.7

he DBSCAN approach offered good results using the cluster validation strategy. For *epsilon* = 0.1 and minPts = 2, it achieved a precision of 90% at thresholds larger than 0.85 for all the similarity measures. The results were even better when the individual validation strategy was applied and the precision reached 90% even with a threshold of 0.75. It can be observed that the precision of the semantic measure is generally very close or higher than that of the vectorial measures above a threshold of 0.7. The intuition that the scores should become more precise as the threshold is raised is proven by the results. Also in a production system, the threshold value can easily be tuned to achieve a desired precision.

The noise produced by DBSCAN was around 30% for *epsilon* = 0.1, meaning if a seller had 99 reviews, 33 of them were discarded because they did not end up in any cluster. The noise increased as *epsilon* was lowered to 0.08 and 0.05. The semantic method outperformed all the others in terms of recall and scored at least double the value of the vectorial measures in the case of the cluster validation strategy for lower thresholds. It scored three times higher than the other measures for the individual review pair validation.



Singleton opinion spam detection

Singleton opinion spam detection - results

OPTICS
minPts=2

Cluster:
P = 75% T > 0.75

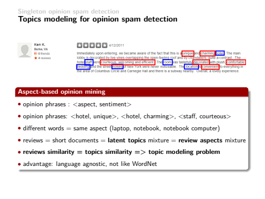
Individual:
P = 80% T > 0.75

Precision is aprox. 10% lower than DBSCAN

OPTICS:
F1 =24%, T = 0.7
DBSCAN:
F1 = 47%, T = 0.7

The OPTICS algorithm produced significantly less noise than DBSCAN because of its ability to adjust to density variations much better. It managed to cluster a very large portion of the dataset and thus more review pairs could be measured for similarity. The precision reached 80%, above a similarity threshold of 0.75, which is 10% lower than for DBSCAN. The recall and F1 scores achieved are also lower. For the semantic measure, it provided a F1 score of only 24% for a precision of 68% at a threshold of 0.7, compared to DBSCAN which got an almost double F1 score of 47% and a precision of 70% for the same threshold.

The semantic measure achieved a better F1 score than the vectorial measures and this proves that semantic similarity outperforms cosine similarity in the detection of deceptive singleton reviews. Generally, the F1 score obtained through the cluster validation strategy is higher than with the individual review pair validation because of the strategy's granularity. All the reviews of the cluster are considered deceptive if at least one review pair goes over the similarity threshold. So, although naturally the first strategy is less precise, it proves that more deceptive reviews surface when a grouping is applied even on straightforward features such as rating and date.



Singleton opinion spam detection

Topics modeling for opinion spam detection

Ken K.
Burke, VA
0 friends
4 reviews

★★★★★ 4/12/2011

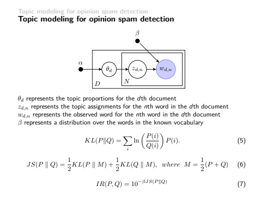
Immediately upon entering, we became aware of the fact that this is a unique and charming hotel. The main lobby is decorated by live vines overlapping the open-feeling roof and by chandaliers, quite a contrast. The hotel staff were courteous, welcoming and efficient. The room was tastefully decorated with plush, comfortable bedding and the street noises of New York were never noticeable. The location is convenient to everything in the area of Columbus Circle and Carnegie Hall and there is a subway nearby. Overall, a lovely experience.

Aspect-based opinion mining

- opinion phrases : <aspect, sentiment>
- opinion phrases: <hotel, unique>, <hotel, charming>, <staff, courteous>
- different words = same aspect (laptop, notebook, notebook computer)
- reviews = short documents = **latent topics** mixture = **review aspects** mixture
- **reviews similarity = topics similarity => topic modeling problem**
- advantage: language agnostic, not like WordNet

It is becoming increasingly difficult to handle the large number of opinions posted on review platforms and at the same time offer this information in a useful way to each user so he or she can make a decision fast whether to buy the product or not. Aspect-based aggregations and short review summaries are used to group and condense what other users think about the product in order to personalize the content served to a new user and shorten the time he needs to make a buying decision.

Aspect mining is a new opinion mining technique used to extract product attributes, also called aspects from reviews. Topic models are statistical models where each document is seen as a mixture of latent topics, each of the topics contributing with certain proportions to the document. Formally, a topic is defined as a distribution over a fixed vocabulary of words.



Topic modeling for opinion spam detection

Topic modeling for opinion spam detection

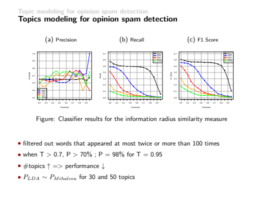
θ_d represents the topic proportions for the *d*th document
$z_{d,n}$ represents the topic assignments for the *n*th word in the *d*th document
$w_{d,n}$ represents the observed word for the *n*th word in the *d*th document
β represents a distribution over the words in the known vocabulary

$$KL(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i). \qquad (5)$$

$$JS(P \parallel Q) = \frac{1}{2}KL(P \parallel M) + \frac{1}{2}KL(Q \parallel M), \;\; where \;\; M = \frac{1}{2}(P+Q) \qquad (6)$$

$$IR(P,Q) = 10^{-\beta JS(P\|Q)} \qquad (7)$$

Each review is a distribution over topics, so computing the similarity between two reviews translates to computing the similarity between their underlying topic distributions. The Kullback-Leibner (KL) measures the difference between two probability distributions *P* and *Q* as shown in equation. So it can be used to compute a value for the distance between the underlying topics distributions of two reviews. This measure has two drawbacks though. If *Q(i)* is zero, then the measure is undefined. It is also not symmetric, meaning the divergence from *P* to *Q* is not the same as that from *Q* to *P*. Translating this to the reviews context, it is not a suitable metric to use, because if a review R_1 is similar to R_2 then it would be expected that R_2 is similar with the same amount to R_1. The Jensen-Shannon (JS) measure is based on the KL divergence and it addresses these drawbacks: it is symmetric and always provides a finite value. It is also bounded by 1, which is more useful when comparing a similarity value for a review pair with a fixed threshold in order to classify the reviews as fake. The JS measure can be rewritten, in order to decrease computational time for large vocabularies, as mentioned by Dagan et al.. IR is short for information radius, while β is a statistical control parameter.



Topic modeling for opinion spam detection

Topics modeling for opinion spam detection

Figure: Classifier results for the information radius similarity measure

- filtered out words that appeared at most twice or more than 100 times
- when T $>$ 0.7, P $>$ 70% ; P $=$ 98% for T $=$ 0.95
- #topics ↑ => performance ↓
- $P_{LDA} \sim P_{Mihalcea}$ for 30 and 50 topics

I filtering out words that appeared either at most twice or more than 100 times in the review corpus. Although the distribution still has a positive skew, the filtering step has managed to improve the similarity results significantly, as it will be shown further on in this section.

The original Blei LDA model was ran on a corpus made up of articles from Science magazine. These are relatively long English texts about various scientific themes, carefully edited to avoid misspelled words. It can be argued that the distribution tail of consumer reviews is longer than in the initial LDA model applied to edited scientific articles, since user reviews are short unedited texts, which can also contain misspelled words. Reviews are also not about varied themes, as the scientific articles, and tend to contain highly frequent words. For 10 topics, the precision is more or less a flat line at 65%. This could happen because 10 topics are way to little to summarize what people say about a seller.