Opinion Spam Detection Through Semantic Similarity

Vlad Sandulescu

DTU Compute
Department of Applied Mathematics and Computer Science

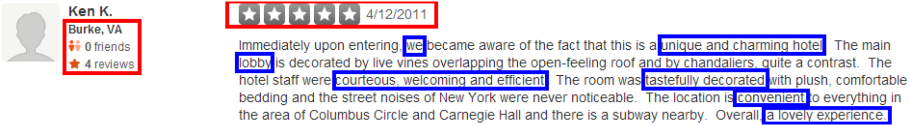
Outline

- Introduction
  - Problem
  - Related Work
  - Goals
  - Textual Similarity - WordNet
- Distribution of truthful and deceptive reviews
- Singleton opinion spam detection
- Topic modeling for opinion spam detection

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

Two main directions: behavioral features and text analysis I



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
- $\sim$ 90% of reviewers write a single review
- 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
- **What happens to singleton reviewers?**

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)

Textual smilarity

Vectorial-based measures

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

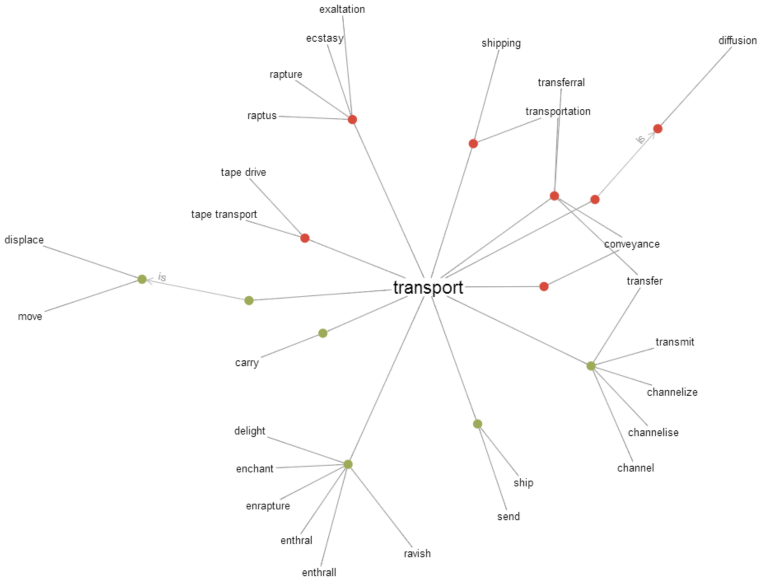
$$\cos(T_1, T_2) = \frac{T_1 T_2}{\|T_1\| \|T_2\|} = \frac{\sum_{i=1}^{n} T_{1i} T_{2i}}{\sqrt{\sum_{i=1}^{n} (T_{1i})^2} \sqrt{\sum_{i=1}^{n} (T_{2i})^2}} \tag{1}$$

Knowledge-based measures

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

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

WordNet synsets



Distribution of truthful and deceptive reviews

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

- Trustpilot dataset - 8990 reviews
- Ott dataset - 800 reviews (400 fake obtained through AMT)
- Stop words removal, POS tagging (extracted NN, JJ, VB)

Cumulative distribution function

$$CDF_X(x) = \mathrm{P}(X \leq x). \tag{3}$$

- Do the distribution curves of the similarity measures overlap for truthful/fake reviews?
- Is there a gap between the two curves? If so, what are its bounds?

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

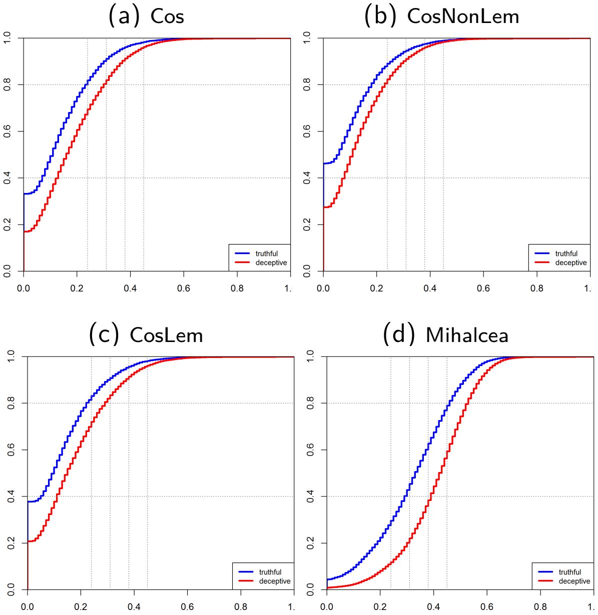
Vectorial ∼ 0.2% diff

- 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



(a) Cos

(b) CosNonLem

(c) CosLem

(d) Mihalcea

Distribution of truthful and deceptive reviews - Ott

Semantic ∼ 8-10% diff

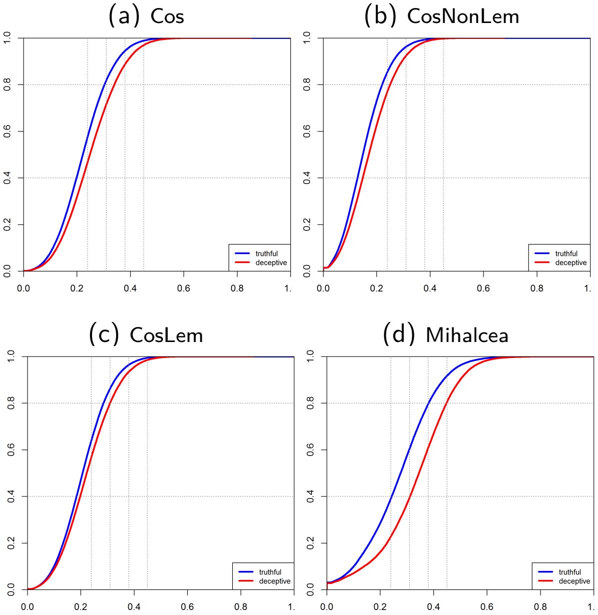
- 40% reviews ↑ 0.22
- 40% reviews ↑ 0.32
- 80% reviews ↑ 0.38
- 80% reviews ↑ 0.44

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

Singleton opinion spam detection - preprocessing & feature design

Trustpilot dataset:

- 8990 English reviews / 130 sellers / 4 or 5 stars / balanced
- min-max normalization => all features in [0, 1]

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{4}$$

- Stanford POS tagger, removed stop words, lemmatization

Behavioral features:

- review title
- review text
- review stars rating
- review date
- user sign up date
- review IP
- proxy IP

"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

Singleton opinion spam detection - clustering

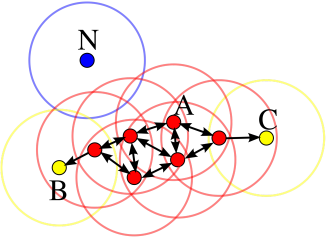
Clustering: DBSCAN & OPTICS

- density reachability
- reduce comparison complexity

$$MaxSimilarity = \max_{R_i, R_j \in C} (sim\_measure(R_i, R_j)) \quad (5)$$

minPts $\in$ *{2,3}*

epsilon $\in$ *{0.05, 0.08, 0.1}*



Similarity measures (all without stop words):

- cosine similarity with all parts-of-speech tags
- cosine similarity with non-lemmatized POSs (NN, VB, JJ)
- cosine similarity with lemmatized POSs (NN, VB, JJ)
- mihalcea semantic similarity (NN, VB, JJ)
- maxsim - maximum value from all

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

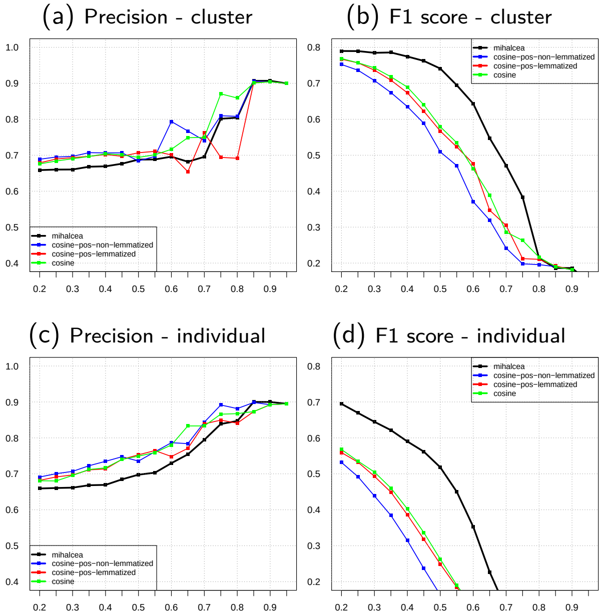
Singleton opinion spam detection - results

DBSCAN
minPts=2
epsilon=0.1

Cluster:
$P = 90\%$ $T > 0.85$

Individual:
$P = 90\%$ $T > 0.75$

semantic $\sim$vectorial
when $T > 0.7$



(a) Precision - cluster

(b) F1 score - cluster

(c) Precision - individual

(d) F1 score - individual

Singleton opinion spam detection - results

DBSCAN

minPts=2

epsilon=0.08

Cluster:

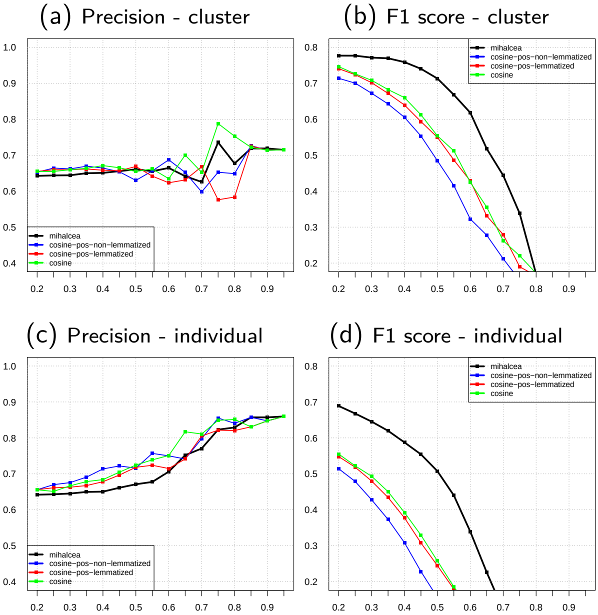
$P = 75\%\ T > 0.75$

Individual:

$P = 75\%\ T > 0.65$

semantic $\sim$vectorial

when $T > 0.7$



(a) Precision - cluster

(b) F1 score - cluster

(c) Precision - individual

(d) F1 score - individual

Singleton opinion spam detection - results

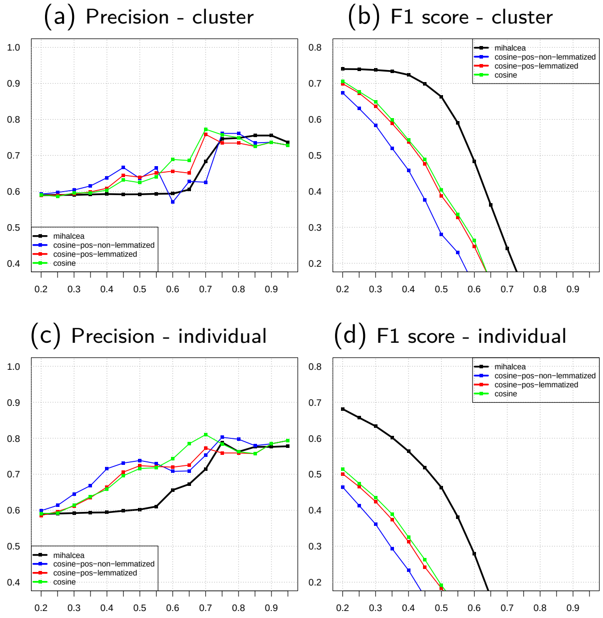
OPTICS
minPts=2

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:
$F1 = 47\%$, $T = 0.7$



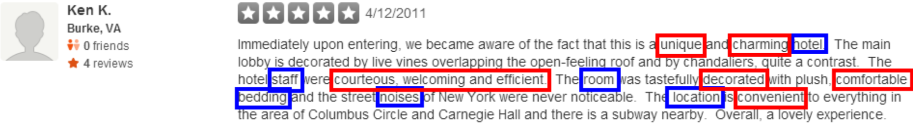
(a) Precision - cluster

(b) F1 score - cluster

(c) Precision - individual

(d) F1 score - individual

Topics modeling for opinion spam detection



Ken K.
Burke, VA
0 friends
4 reviews

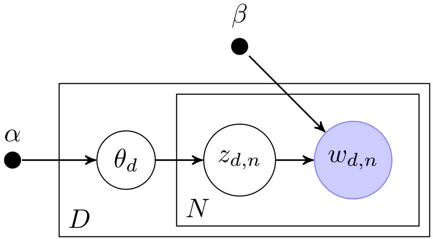
★★★★★ 4/12/2011

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

Topic modeling for opinion spam detection



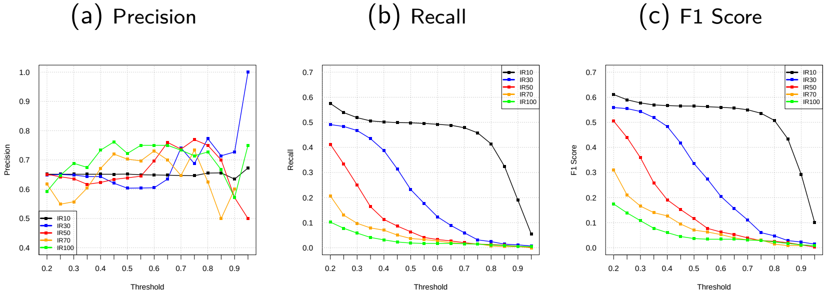
θ_d represents the topic proportions for the *d*th document
$z_{d,n}$ represents the topic assignments for the *n*th word in the *d*th document
$w_{d,n}$ represents the observed word for the *n*th word in the *d*th document
β represents a distribution over the words in the known vocabulary

$$KL(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i). \tag{6}$$

$$JS(P \parallel Q) = \frac{1}{2}KL(P \parallel M) + \frac{1}{2}KL(Q \parallel M), \;\; where \;\; M = \frac{1}{2}(P+Q) \tag{7}$$

$$IR(P,Q) = 10^{-\beta JS(P\|Q)} \tag{8}$$

Topics modeling for opinion spam detection


(a) Precision

(b) Recall

(c) F1 Score

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

Wrap-up

Key points

- shape of reviews in Trustpilot and Ott datasets => semantic similarity shows a more distinctive gap
- opinion spam detection using two new methods
- semantic similarity with WordNet => can outperform the vectorial-based measures
- topic modeling with LDA => performance ∼ vectorial models
- density clustering with DBSCAN and OPTICS on behavioral features
- comparison with cosine similarity and variations
- precision is good enough for production systems

Questions?