Synthetic Review Spamming and Defense

Huan Sun, Alex Morales, and Xifeng Yan
Department of Computer Science
University of California, Santa Barbara

Introduction

- We propose a simple, but powerful review spamming technique to automatically synthesize reviews;
- This kind of review spam is hard to detect: Both the state-of-the-art computational approaches and human readers acquire an error rate of 35%-48%;
- We propose a novel defense method based on the differences of semantic flows between truthful reviews and synthesized reviews, which significantly reduces the detection error rate by approximately 14%;
- Through this study, we hope to stimulate debate and defense against the machine-synthesized fake reviews.

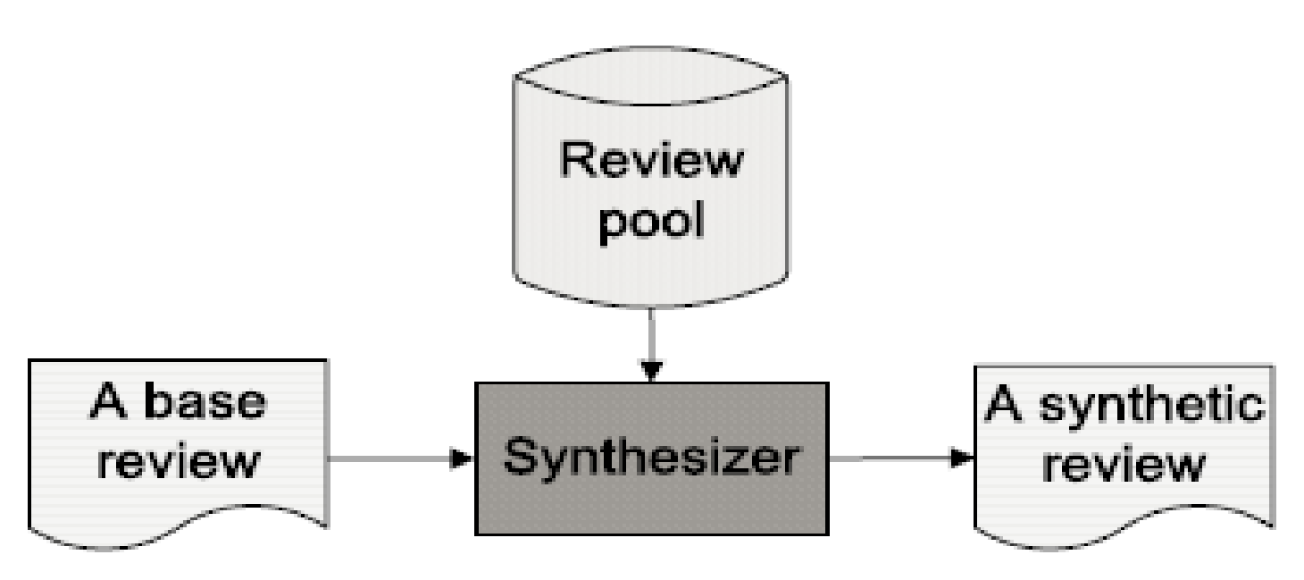
Welcome to try our demo:
www.cs.ucsb.edu/~alex_morales/reviewspam/

Automatic Review Generation

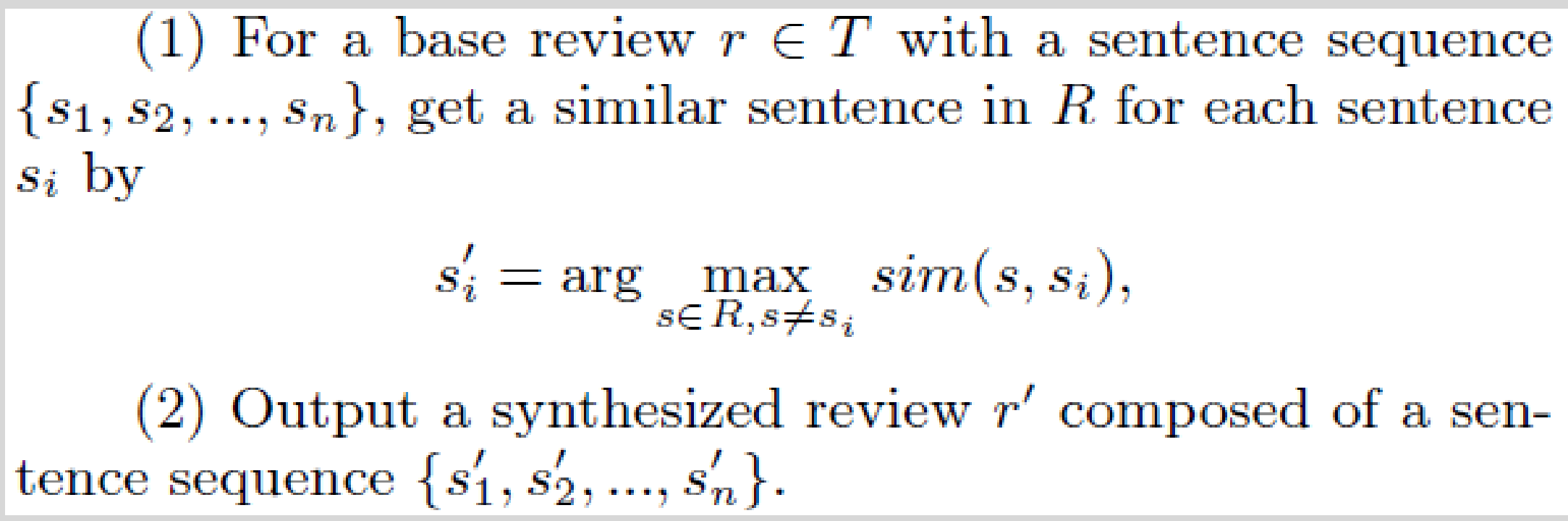
Motivation

- Hiring humans to write reviews can be expensive both in time and cost;
- Synthesizing reviews automatically is low-cost, high-throughput, and can be (or has been) employed by evil attackers. Moreover, current fake detection algorithms achieve bad performance on detecting such synthetic reviews.

Review Synthesis Model



[Review pool]: truthful reviews from online websites like TripAdvisor; short reviews that are not content rich are discarded.
[Base review]: A base review is randomly drawn from the pool, based on which a synthetic review will be generated.
[Synthesizer]: The synthesizer replaces each sentence in a base review by the most similar (not exactly the same) sentence in the review pool; i.e.,



(1) For a base review $r \in T$ with a sentence sequence $\{s_1, s_2, ..., s_n\}$, get a similar sentence in R for each sentence s_i by

$$s'_i = \arg \max_{s \in R, s \neq s_i} sim(s, s_i),$$

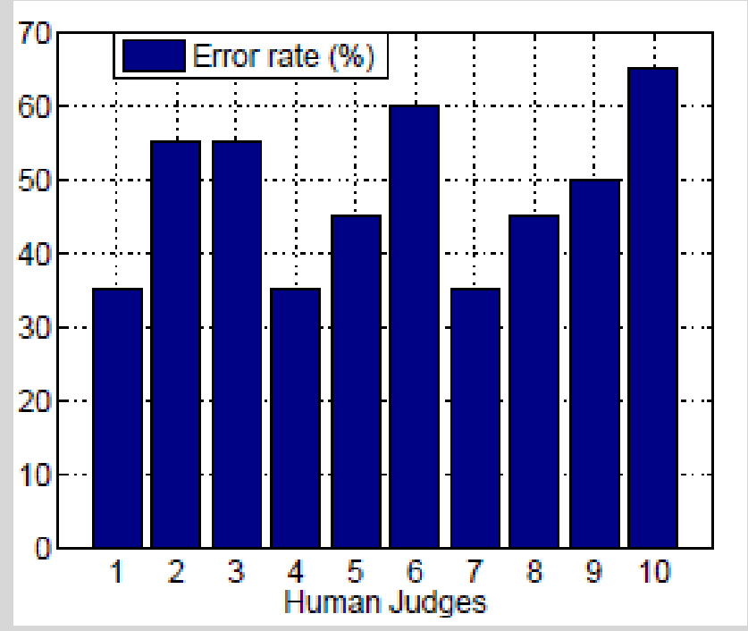
(2) Output a synthesized review r' composed of a sentence sequence $\{s'_1, s'_2, ..., s'_n\}$.

Performance of Existing Detectors

Human readers: with an error rate of 48% on average;

Automatic Review Generation (Cont'd)

State-of-the-art detectors: with an error rate of 34%-44% on average.



Algorithms	Error rate
Ott et al. [1]	40.5%
Liu et al. [2]	34.5%
Harris et al. [3]	43.3%

Welcome to test your detection ability in our demo website:
www.cs.ucsb.edu/~alex_morales/reviewspam/

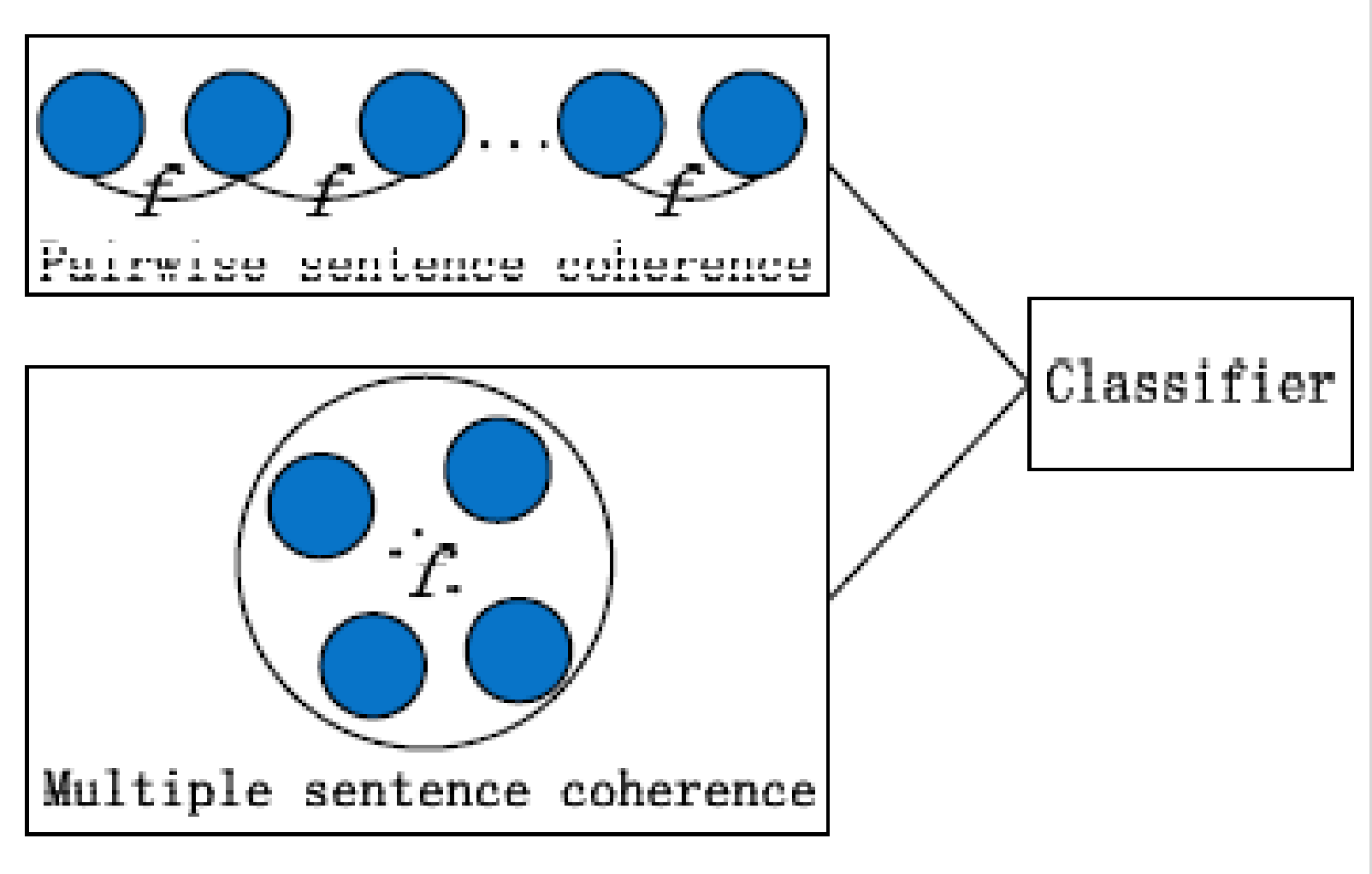
Discussions on Review Synthesis Model

- **Why bad detection performance?** The base reviews, indeed written by humans, provide a good skeleton and make the synthetic reviews as authentic-looking as possible;
- **Sentence duplication concern?** To pass sentence-level duplication detectors, one can further employ sentence-rewriting/paraphrase techniques, which are not our focus here.
- **Local text duplication?** Local text content (e.g., n-grams of an article) is not a unique fingerprint of one specific review. Local text duplication detectors can incur a high false positive rate.

Synthetic Review Detection

Synthetic reviews using sentence transplants should bear subtle semantic incoherence between sentences. We propose a general methodology for coherence analysis.

A General Methodology



Pairwise sentence coherence: to measure the coherence between sentences.
Multiple sentence coherence: to measure the stretch and changes in multiple consecutive sentences.
Such two kinds of features are extracted for classification.

Our Instantiation of the Methodology

- Sentence transition

In natural human writings, we expect to see certain words in the current sentence, given the words we observed in the previous sentence. Word to word transition probabilities are named pointwise transition probability (PTP).

$$\mathrm{PTP}_{(i,:)} = [P(\omega_1 \mid \omega_i), ..., P(\omega_j \mid \omega_i), ..., P(\omega_n \mid \omega_i)]$$
$$s.t., \sum_{\omega_j \in W} P(\omega_j \mid \omega_i) = 1$$

Synthetic Review Detection (Cont'd)

We calculate the sentence transition probability as:

$$\mathrm{P}(\omega_j \mid \mathrm{s}_1) = \sum_{\omega_i \in s_1} \theta(\omega_i, \mathrm{s}_1)\, \mathrm{P}(\omega_j \mid \omega_i)$$
$$P(s_1 \to s_2) = \prod_{\omega_j \in s_2} P(\omega_j \mid s_1)^{c(\omega_j, s_2)}$$
$$\theta(\omega_i, \mathrm{s}_1) = \frac{c(\omega_i, s_1)}{|s_1|}$$

Other alternatives are also tested.
Perplexity as a measure of sentence transitions in a review:

$$\mathrm{Perplexity}(r) = \exp\left(-\frac{\sum_{i=1}^{n-1} \log(P(s_i \to s_{i+1}))}{\sum_{i=1}^{n-1} |s_{i+1}|}\right)$$

- Word co-occurrence

Words tend to demonstrate co-occurrence patterns in consecutive sentences. The co-occurrence score for two words:

$$O_{i,j} = \log\left(\frac{P_{i,j}}{P_i P_j}\right)$$

Sentence co-occurrence (SCO) score:

$$\mathrm{SCO}(s_1, s_2) = \frac{1}{|s_1||s_2|} \sum_{\omega_i \in s_1, \omega_j \in s_2} O_{i,j}$$

- Pairwise-sentence similarity

Sentence similarities based on word overlap, wordnet, and LSI are taken into account. The average of scores for all the sentence pairs in a review is used as one measure, denoted as SIM.

- Multiple sentence coherence

Semantic dispersion (SD) to measure how dispersed/focused a review is:

$$\mathrm{SD} = \frac{1}{n} \sum_{i=1}^{n} \| \upsilon_i - \mathrm{centroid} \|$$

Each sentence is represented as a semantic vector given by LSI. Centroid is the average of all the sentence vectors in a review. A *running length* measure for occasional semantic jumps between adjacent sentences is also tested.

Experiments

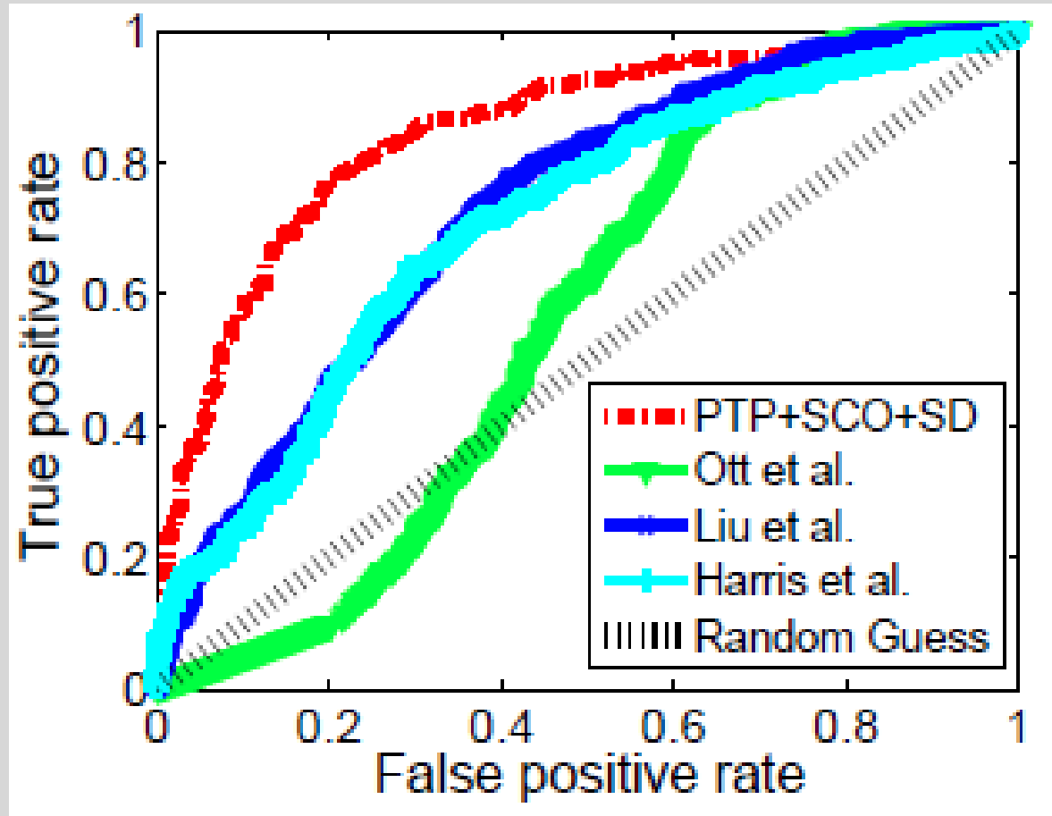
- **Datasets**: 10 datasets; each has 500/500 truthful/fake reviews;
- **Classification**: three popular classifiers for classification: Support Vector Machine with linear Kernel (Linear), polynomial Kernel (Poly), and Naïve Bayes (NB); 5-fold cross-validation;
- **Performance measure**: error rate, i.e., the number of misclassified reviews to the total number of reviews.

Error Rate Comparison

Instantiated measures	Error Rate (%)		
	Linear	Poly	NB
PTP	38.6	41.2	36.7
SCO	39.0	39.0	38.7
SIM	48.0-49.8	48.7-50.3	47.9-52.2
SD	41.5	39.3	32.7
PTP+SCO+SD	21.6	22.0	24.5
Ott et al. [1]	40.5		
Liu et al. [2]	34.5		
Harris et al. [3]	43.3		
Human	48.0		

Experiments (Cont'd)

ROC Comparison

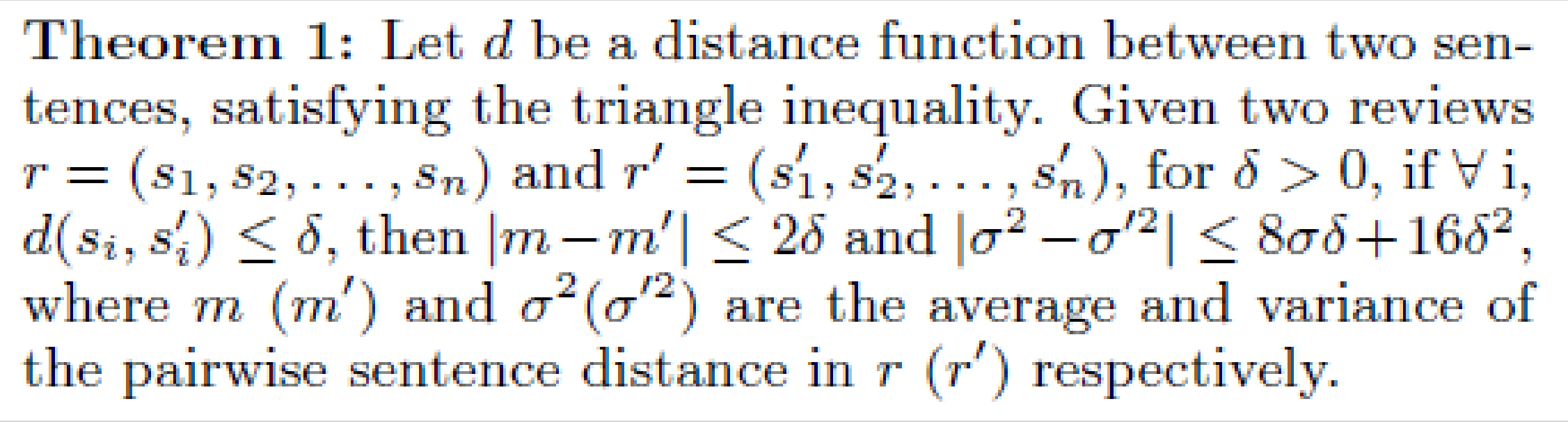


Adaptability Study

Algorithms	Error rate
Ott et al. [1]	39.6%
Liu et al. [2]	41.3%
Harris et al. [3]	43.4%
Our Method	26.7%

In adaptability study, we use reviews from different cities respectively for training and testing.

Towards Understanding Why SIM Performs Badly



Theorem 1: Let d be a distance function between two sentences, satisfying the triangle inequality. Given two reviews $r = (s_1, s_2, \ldots, s_n)$ and $r' = (s'_1, s'_2, \ldots, s'_n)$, for $\delta > 0$, if $\forall$ i, $d(s_i, s'_i) \leq \delta$, then $|m - m'| \leq 2\delta$ and $|\sigma^2 - \sigma'^2| \leq 8\sigma\delta + 16\delta^2$, where m (m') and $\sigma^2(\sigma'^2)$ are the average and variance of the pairwise sentence distance in r (r') respectively.

Since the synthesis procedure uses similar sentences to generate a review, it tends to make the SIM measures of truthful reviews quite close to those of synthetic ones.

Ranking Reviews Based on Authenticity

Precision@K: the ratio of truthful ones in the top K reviews.

Precision@	PTP+SCO+SD	Ott et al. [1]	Liu et al. [2]	Harris et al. [3]
20	0.98	0.62	0.72	0.82
50	0.97	0.47	0.79	0.84
100	0.96	0.47	0.80	0.82
200	0.93	0.52	0.78	0.77

Conclusion

- We introduce a simple, yet powerful review spamming technique that could fail the existing detection algorithms easily;
- The instantiated framework with our new coherence measures can significantly improve the detection performance by roughly 14%;
- It is still an open research problem to further improve the detection accuracy. One meaningful extension is to study the prevalence of synthetic reviews in real review environment.

References

[1] Ott, Myle, *et al*. "Finding deceptive opinion spam by any stretch of the imagination." *ACL*, 2011.
[2] Liu, Jingjing, *et al*. "Low-quality product review detection in opinion summarization." *EMNLP-CoNLL*, 2007.
[3] Harris, Christopher G. "Detecting Deceptive Opinion Spam Using Human Computation." *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.