ICWSM – A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews

Oren Tsur
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
oren@cs.huji.ac.il

Dmitry Davidov
ICNC
The Hebrew University
Jerusalem, Israel
dmitry@alice.nc.huji.ac.il

Ari Rappoport
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
www.cs.huji.ac.il/∼arir

Abstract

Sarcasm is a sophisticated form of speech act widely used in online communities. Automatic recognition of sarcasm is, however, a novel task. Sarcasm recognition could contribute to the performance of review summarization and ranking systems. This paper presents SASI, a novel Semi-supervised Algorithm for Sarcasm Identification that recognizes sarcastic sentences in product reviews. SASI has two stages: semi-supervised pattern acquisition, and sarcasm classification.

We experimented on a data set of about 66000 Amazon reviews for various books and products. Using a gold standard in which each sentence was tagged by 3 annotators, we obtained precision of 77% and recall of 83.1% for identifying sarcastic sentences. We found some strong features that characterize sarcastic utterances. However, a combination of more subtle pattern-based features proved more promising in identifying the various facets of sarcasm. We also speculate on the motivation for using sarcasm in online communities and social networks.

Introduction

Indirect speech is a sophisticated form of speech act in which speakers convey their message in an implicit way. One manifestation of indirect speech acts is *sarcasm* (or *verbal irony*). Sarcastic writing is common in opinionated user generated content such as blog posts and product reviews. The inherently ambiguous nature of sarcasm sometimes makes it hard even for humans to decide whether an utterance is sarcastic or not. In this paper we present a novel algorithm for automatic identification of sarcastic sentences in product reviews.

One definition for sarcasm is: *the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry* (Macmillan English Dictionary 2007). While this definition holds in many cases, sarcasm manifests itself in many other ways (Brown 1980; Gibbs and O'Brien 1991). It is best to present a number of examples which show different facets of the phenomenon. The following sentences are all review titles (summaries), taken from our experimental data set:

1. *"[I] Love The Cover" (book)*
2. *"Where am I?" (GPS device)*
3. *"Trees died for this book?" (book)*
4. *"Be sure to save your purchase receipt" (smart phone)*
5. *"Are these iPods designed to die after two years?" (music player)*
6. *"Great for insomniacs" (book)*
7. *"All the features you want. Too bad they don't work!" (smart phone)*
8. *"Great idea, now try again with a real product development team" (e-reader)*
9. *"Defective by design" (music player)*

It would not be appropriate to discuss each example in detail here, so we outline some important observations. Example (1) might be a genuine complement if it appears in the body of the review. However, recalling the expression 'don't judge a book by its cover' and choosing it as the title of the review reveals its sarcastic nature. While (2) requires the knowledge of the context (review of a GPS device), (3) is sarcastic independently of context. (4) might seem as the borderline between suggesting a good practice and a sarcastic utterance, however, like (1), placing it as the title of the review leaves no doubts regarding its sarcastic meaning. In (5) the sarcasm emerges from the naive-like question phrasing that assumes the general expectation that goods should last. In (6) the sarcasm requires world knowledge (insomnia vs. boredom ↦ sleep) and in (7,8) the sarcasm is conveyed by the explicit contradiction. Interestingly, (8) contains an explicit positive sentiment ('great idea') while the positive sentiment in (7) doesn't make use of an explicit sentiment word. Although the negative sentiment is very explicit in the iPod review (9), the sarcastic effect emerges from the pun that assumes the knowledge that the design is one of the most celebrated features of Apple's products. It is important to mention that none of the above reasoning was directly introduced to our algorithm. This will be further addressed in the algorithm overview and in the discussion sections.

Beyond the obvious psychology and cognitive science interest in suggesting models for the use and recognition of sarcasm, automatic detection of sarcasm is interesting from a commercial point of view. Studies of user preferences suggest that some users find sarcastic reviews biased and less helpful while others prefer reading sarcastic reviews (the 'brilliant-but-cruel' hypothesis (Danescu-Niculescu-Mizil et al. 2009)). Identification of sarcastic reviews can therefore improve the *personalization* of content ranking and recommendation systems such as (Tsur and Rappoport 2009).

Another important benefit is the improvement of review summarization and opinion mining systems such as (Popescu and Etzioni 2005; Pang and Lee 2004; Wiebe et al. 2004; Hu and Liu 2004; Kessler and Nicolov 2009), currently incapable of dealing with sarcastic sentences. Typically, these systems employ three main steps: (1) feature identification, (2) sentiment analysis, and (3) averaging the sentiment score for each feature. Sarcasm, at its core, may harm opinion mining systems since its explicit meaning is different or opposite from the real intended meaning (see examples 6-8), thus averaging on the sentiment would not be accurate.

In this paper we present SASI, a novel Semi-supervised Algorithm for Sarcasm Identification. The algorithm employs two modules: (I) semi supervised pattern acquisition for identifying sarcastic patterns that serve as features for a classifier, and (II) a classification algorithm that classifies each sentence to a sarcastic class.

We evaluated our system on a large collection of Amazon.com user reviews for different types of products, showing good results and substantially outperforming a strong baseline based on sentiment.

The paper is arranged as follows. The next section surveys relevant work and outlines the theoretical framework. The third section presents the pattern acquisition algorithm and the classification algorithm. Section 4 presents the experimental setup and the evaluation procedure. Results are presented in the following section, followed by a short discussion.

Related Work

While the use of irony and sarcasm is well studied from its linguistic and psychologic aspects (Muecke 1982; Stingfellow 1994; Gibbs and Colston 2007), automatic recognition of sarcasm is a novel task in natural language processing, and only few works address the issue. In computational works, mainly on sentiment analysis, sarcasm is mentioned briefly as a hard nut that is yet to be cracked. For a comprehensive overview of the state of the art and challenges of opinion mining and sentiment analysis see Pang and Lee (2008).

Tepperman et al. (2006) identify sarcasm in spoken dialogue systems, however, their work is restricted to sarcastic utterances that contain the expression 'yeah-right' and they depend heavily on cues in the spoken dialogue such as laughter, pauses within the speech stream, the gender (recognized by voice) of the speaker and some prosodic features.

Burfoot and Baldwin (2009) use SVM to determine whether newswire articles are true or satirical. They introduce the notion of *validity* which models absurdity via a measure somewhat close to PMI. Validity is relatively lower when a sentence include a made-up entity or when a sentence contains unusual combinations of named entities such as, for example, those in the satirical article beginning "Missing Brazilian balloonist Padre spotted straddling Pink Floyd flying pig". We note that while sarcasm can be based on exaggeration or unusual collocations, this model covers only a limited subset of the sarcastic utterances.

Utsumi (1996; 2000) introduces the *implicit display* theory, a cognitive computational framework that models the *ironic environment*. The complex axiomatic system depends heavily on world knowledge ('universal' or 'common' knowledge in AI terms) and expectations. It requires a thorough analysis of each utterance and its context to match predicates in a specific logical formalism. While comprehensive, it is currently impractical to implement on a large scale or for an open domain.

Polanti and Zaenen (2006) suggest a theoretical framework in which the context of sentiment words shifts the valence of the expressed sentiment.

Mihalcea and Strapparava (2005) and Mihalcea and Pulman (2007) present a system that identifies humorous one-liners. They classify sentences using naive Bayes and SVM. They conclude that the most frequently observed semantic features are negative polarity and human-centeredness.

Some philosophical, psychological and linguistic theories of irony and sarcasm are worth referencing as a theoretical framework: the *constraints satisfaction* theory (Utsumi 1996; Katz 2005), the *role playing* theory (Clark and Gerrig 1984), the *echoic mention* framework (Wilson and Sperber 1992) and *pretence* framework (Gibbs 1986), all based on violation of the maxims proposed by Grice (1975).

Classification Framework and Algorithm

Our sarcasm classification method is based on the classic semi-supervised learning framework. For the training phase, we were given a small set of manually labeled sentences (seeds). A discrete score $1 \dots 5$ was assigned to each sentence in the training set, where five means a definitely sarcastic sentence and one means a clear absence of sarcasm.

Given the labeled sentences, we extracted a set of features to be used in feature vectors. We utilized two basic feature types: syntactic and pattern-based features. In order to overcome the sparsity of sarcastic sentences and to avoid noisy and labor intensive wide scale annotation, we executed search engine queries in order to acquire more examples and automatically expand the training set. We then constructed feature vectors for each of the labeled examples in the expanded training set and used them to build the model and assign scores to unlabeled examples. The remainder of this section provides a detailed description of the algorithm.

Preprocessing of data

Each review is usually focused on some specific company/author and its product/book. The name of this product/author usually appears many times in the review text. Since our main feature type is surface patterns, we would like to capture helpful patterns which include such names. However, we would like to avoid extraction of author-specific or product-specific patterns which are only useful for specific product or company.

In order to produce less specific patterns, we automatically replace each appearance of a product/author/company/book name with corresponding generalized '[product]','[company]','[title]' and '[author]' tags[1]. We also removed all HTML tags and special symbols from the review text.

Pattern-based features

Pattern extraction Our main feature type is based on surface patterns. In order to extract such patterns automatically, we followed the algorithm given in (Davidov and Rappoport 2006). We classified words into high-frequency words (HFWs) and content words (CWs). A word whose corpus frequency is more (less) than F_H (F_C) is considered to be a HFW (CW). Unlike (Davidov and Rappoport 2006), we consider all punctuation characters as HFWs. We also consider [product], [company], [title], [author] tags as HFWs for pattern extraction. We define a pattern as an ordered sequence of high frequency words and slots for content words. Following (Davidov and Rappoport 2008) F_H and F_C thresholds were set to 1000 words per million (upper bound for F_C) and 100 words per million (lower bound for F_H)[2].

In our patterns we allow 2-6 HFWs and 1-6 slots for CWs. To avoid collection of patterns which capture a part of a multiword expression, we require patterns to start and to end with a HFW. Thus a minimal pattern is of the form [HFW] [CW slot] [HFW]. For each sentence it is possible to generate dozens of patterns that may overlap.

For example, given a sentence "Garmin apparently does not care much about product quality or customer support", we have generated several patterns including "[company] CW does not CW much", "does not CW much about CW CW or", "not CW much" and "about CW CW or CW CW.". Note that "[company]" and "." are treated as high frequency words.

Pattern selection The first stage provided us with hundreds of patterns. However, only some of them are useful since many of them are either too general or too specific. In order to reduce the feature space, we have used two criteria to select useful patterns. First, we removed all patterns which appear only in sentences originating from a single product/book. Such patterns are usually product-specific like "looking for a CW camera" (e.g., where the CW is 'Sony').

Next we removed all patterns which appear in the training set both in some example labeled 5 (clearly sarcastic) and in some other example labeled 1 (obviously not sarcastic). This way we filter out general and frequent patterns like 'either CW or CW.". Such patterns are usually too generic and uninformative for our task.

Pattern matching Once patterns are selected, we have used each pattern to construct a single entry in the feature vectors. For each sentence we calculated feature value for each pattern as following:

$$\begin{cases} 1: & \text{Exact match -- all the pattern components} \\ & \text{appear in the sentence in correct} \\ & \text{order without any additional words.} \\ \alpha: & \text{Sparse match -- same as exact match} \\ & \text{but additional non-matching words can be} \\ & \text{inserted between pattern components.} \\ \gamma * n/N: & \text{Incomplete match -- only } n > 1 \text{ of } N \text{ pattern} \\ & \text{components appear in the sentence,} \\ & \text{while some non-matching words can} \\ & \text{be inserted in-between. At least one of the} \\ & \text{appearing components should be a HFW.} \\ 0: & \text{No match -- nothing or only a single} \\ & \text{pattern component appears in the sentence.} \end{cases}$$

$0 \le \alpha \le 1$ and $0 \le \gamma \le 1$ are parameters we use to assign reduced scores for imperfect matches. Since the patterns we use are relatively long, exact matches are uncommon, and taking advantage of partial matches allows us to significantly reduce the sparsity of the feature vectors. We used $\alpha = \gamma = 0.1$ in all experiments.

Thus, for the sentence "Garmin apparently does not care much about product quality or customer support", the value for "[title] CW does not" would be 1 (exact match); for "[title] CW not" would be 0.1 (sparse match due to insertion of 'does'); and for "[title] CW CW does not" would be $0.1 * 4/5 = 0.08$ (incomplete match since the second CW is missing).

Punctuation-based features

In addition to pattern-based features we have used the following generic features. All these features were normalized to be in [0-1] by dividing them by the maximal observed value, thus the weight of each of these features is equal to the weight of a single pattern feature.

1. Sentence length in words.
2. Number of "!" characters in the sentence.
3. Number of "?" characters in the sentence.
4. Number of quotes in the sentence.
5. Number of capitalized/all capitals words in the sentence.

Data enrichment

Since we start with only a small annotated seed for training (and particularly the number of clearly sarcastic sentences in the seed is relatively modest) and since annotation is noisy and expensive, we would like to find more training examples without requiring additional annotation effort.

To achieve this, we posited that sarcastic sentences frequently co-appear in texts with other sarcastic sentences.

[1] We assume that appropriate names are provided with each review, which is a reasonable assumption for the Amazon reviews.

[2] Note that F_H and F_C set bounds that allow overlap between some HFWs and CWs. See (Davidov and Rappoport 2008) for a short discussion.

We performed an automated web search using the Yahoo! BOSS API[3], where for each sentence s in the training set (seed), we composed a search engine query q_s containing this sentence[4]. We collected up to 50 search engine snippets for each example and added the sentences found in these snippets to the training set. The label (level of sarcasm) $Label(s_q)$ of a newly extracted sentence s_q is similar to the label $Label(s)$ of the seed sentence s that was used for the query that acquired it.

The seed sentences together with newly acquired sentences constitutes the (enriched) training set.

Here are two examples. For a training sarcastic sentence "This book was really good-until page 2!", the framework would execute the query "this book was really good until", retrieving both different sarcastic sentences which include these 6 words ("Gee, I thought this book was really good until I found out the author didn't get into Bread Loaf!") and accompanying snippet sentences such as "It just didn't make much sense.". Similarly, for a training sentence "I guess I am not intellectual enough to get into this novel", the query string is "I guess I am not intellectual", a similar sentence retrieved is "I guess I am not intellectual enough to understand it", and an accompanied sentence is "It reads more like a journal than a novel".

Classification

In order to assign a score to new examples in the test set we use a k-nearest neighbors (kNN)-like strategy. We construct feature vectors for each example in the training and test sets. We would like to calculate the score for each example in the test set. For each feature vector v in the test set, we compute the Euclidean distance to each of the matching vectors in the extended training set, where matching vectors are defined as ones which share at least one pattern feature with v.

Let $t_i, i = 1..k$ be the k vectors with lowest Euclidean distance to v[5]. Then v is classified with a label l as follows:

$$Count(l) = \text{Fraction of vectors in the training set with label } l$$

$$Label(v) = \left[\frac{1}{k} \sum_i \frac{Count(Label(t_i)) \cdot Label(t_i)}{\sum_j Count(label(t_j))} \right]$$

Thus the score is a weighted average of the k closest training set vectors. If there are less than k matching vectors for the given example then fewer vectors are used in the computation. If there are no matching vectors found for v, we assigned the default value $Label(v) = 1$ (not sarcastic at all), since sarcastic sentences are fewer in number than non-sarcastic ones (this is a 'most common tag' strategy).

Baseline

A common baseline can be 'pick the majority class', however, since sarcastic sentences are sparse, this will obviously achieve good precision (computed over all sentences) but close to zero recall. The sparsity of sarcastic sentences was also proved in our manual seed annotation. Instead, we propose a stronger heuristic baseline[6].

#products	**#reviews**	**avg. stars**	**avg. length (chars)**
120	66271	4.19	953

Table 1: Corpus statistics.

Star-sentiment baseline Many studies on sarcasm suggest that sarcasm emerges from the gap between the expected utterance and the actual utterance exaggeration and overstatement, as modeled in the echoic mention, allusion and pretense theories (see Related Work section). We implemented a strong baseline designed to capture the notion of sarcasm as reflected by those models, and trying to meet the definition "saying the opposite of what you mean in a way intended to make someone else feel stupid or show you are angry" (Macmillan 2007).

We exploit the meta-data provided by Amazon, namely the star rating each reviewer is obliged to provide, in order to identify unhappy reviewers (reviews with 1-3 stars i.e. the review presented at Table 1). From this set of negative reviews, our baseline classifies as sarcastic those sentences that exhibit strong positive sentiment. The list of positive sentiment words is predefined and captures words typically found in reviews (for example, 'great', 'excellent', 'best', 'top', 'exciting', etc), about twenty words in total. This baseline is a high quality one as it is manually tailored to capture the main characteristics of sarcasm as accepted by the linguistic and psychological communities.

Data and Evaluation Setup

Data

We are interested in identification of sarcastic sentences in online product reviews. For our experiments we used a collection of 66000 reviews for 120 products extracted from Amazon.com. The collection contained reviews for products from very different domains: books (fiction, non fiction, children), music players, digital cameras, camcoders, GPS devices, e-readers, game consoles, mobile phones and more. Some more details about the data are summarized in Table 1. Figure 1 illustrates the structure of a typical review.

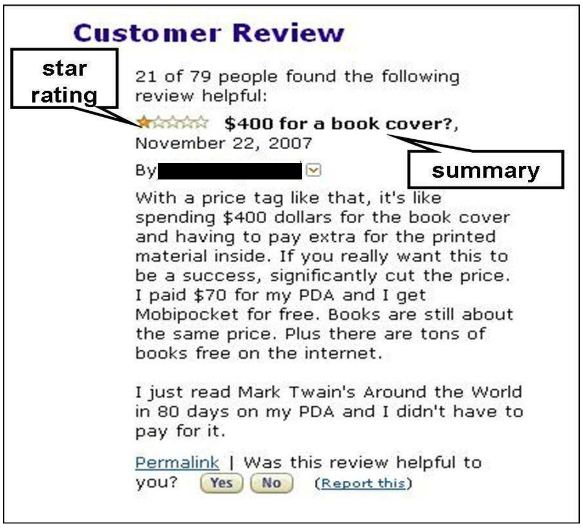
Seed training set. As described in the previous section, SASI is semi supervised, hence requires a small seed of annotated data. The seed consisted of 80 sentences from the corpus which were manually labeled as sarcastic to some degree (labels 3-5) and of the full text of 80 negative reviews that found to contain no sarcastic sentences. These included 505 sentences that are clearly not sarcastic as negative examples.

[3] http://developer.yahoo.com/search/boss.

[4] If the sentence contained more than 6 words, only the first 6 words were included in the search engine query.

[5] We used $k = 5$ for all experiments.

[6] We note that sarcasm annotation is extremely expensive due to the sparseness of sarcastic utterances, hence, no supervised baseline is available. Moreover, we took the semi-supervised approach in order to overcome the need for expensive annotation. However, results are evaluated against an ensemble of human annotators.



Customer Review

21 of 79 people found the following review helpful:

$400 for a book cover?, November 22, 2007

By [redacted]

With a price tag like that, it's like spending $400 dollars for the book cover and having to pay extra for the printed material inside. If you really want this to be a success, significantly cut the price. I paid $70 for my PDA and I get Mobipocket for free. Books are still about the same price. Plus there are tons of books free on the internet.

I just read Mark Twain's Around the World in 80 days on my PDA and I didn't have to pay for it.

Permalink | Was this review helpful to you? (Report this)

Figure 1: A screen shot of an amazon review for the kindle eReader. A reviewer needs to provide three information types: *star rating* (1-5), a one sentence *summary*, and the *body* of the review.

Extended training set. After expanding the training set, our training data now contains 471 positive examples and 5020 negative examples. This ratio is to be expected, since non-sarcastic sentences outnumber sarcastic ones. In addition, sarcastic sentences are usually present in negative reviews, while most online reviews are positive (Liu et al. 2007). This general tendency to positivity also reflects in our data, as can be seen from the average number of stars in Table 1.

Evaluation procedure

We used two experimental frameworks to test SASI's accuracy. In the first experiment we evaluated the pattern acquisition process, how consistent it is and to what extent it contributes to correct classification. We did that by 5-fold cross validation over the seed data. In the second experiment we evaluated SASI on a test set of unseen sentences, comparing its output to a gold standard annotated by a large number of human annotators. This way we verify that there is no over-fitting and that the algorithm is not biased by the notion of sarcasm of a single seed annotator.

5-fold cross validation. In this experimental setting, the seed data was divided to 5 parts and a 5-fold cross validation test is executed. Each time, we use 4 parts of the seed as the training data and only this part is used for the feature selection and data enrichment. This 5-fold process was repeated ten times. In order to learn about the contribution of every feature type, we repeated this experiment several more times with different sets of optional features.

We used 5-fold cross validation and not the standard 10-fold since the number of seed examples (especially positive) is relatively small hence 10-fold is too sensitive to noise.

Classifying new sentences. Evaluation of sarcasm is a hard task due to the elusive nature of sarcasm, as discussed in the Introduction. The subtleties of sarcasm are context sensitive, culturally dependent and generally fuzzy. In order to evaluate the quaity of our algorithm, we used SASI to classify all sentences in the corpus of 66000 reviews (besides the small seed that was pre-annotated and was used for the evaluation in the 5-fold cross validation experiment). Since it is impossible to create a gold standard classification of each and every sentence in the corpus, we created a small test set by sampling 90 sentences which were classified as sarcastic (labels 3-5) and 90 sentences classified as not sarcastic (labels 1,2).

In order to make the evaluation fair (harsher) and more relevant, we introduced two constraints to the sampling process. First, we restricted the non-sarcastic sentences to belong to negative reviews (1-3 stars) so that all sentences in the evaluation set are drawn from the same population, increasing the chances they convey various levels of direct or indirect *negative* sentiment.

This constraint makes evaluation harsher on our algorithm since the evaluation set is expected to contain different types of non-sarcastic negative sentiment sentences, in addition to non-trivial sarcastic sentences that do not necessarily obey to the "saying the opposite" definition (these are nicely captured by our baseline).

Second, we sampled only sentences containing a named-entity or a reference to a named entity. This constraint was introduced in order to keep the evaluation set relevant, since sentences that refer to the named entity (product/ manufacturer/ title/ author) are more likely to contain an explicit or implicit sentiment.

Procedure The evaluation set was randomly divided to 5 batches. Each batch contained 36 sentences from the evaluation set and 4 anchor sentences:

1. *"I love it, although i should have waited 2 more weeks for the touch or the classic."*
2. *"Horrible tripe of a novel, i Lost IQ points reading it"*
3. *"All the features you want – too bad they don't work!"*
4. *"Enjoyable light holiday reading."*

Anchors 1 and 4 are non-sarcastic and 2 and 3 are sarcastic. The anchor sentences were not part of the test set and were the same in all five batches. The purpose of the anchor sentences is to control the evaluation procedure and verify that annotators are not assigning sarcastic labels randomly. Obviously, we ignored the anchor sentences when assessing the algorithm's accuracy.

In order to create a gold standard we employed 15 adult annotators of varying cultural backgrounds, all fluent English speakers, accustomed to reading product reviews on Amazon. We used a relatively large number of annotators in order to overcome the possible bias induced by personal character and ethnicity/culture of a single annotator (Muecke 1982). Each annotator was asked to assess the level of sarcasm of each sentence of a set of 40 sentences on a scale of 1-5.

In total, each sentence was annotated by three different annotators.

Inter Annotator Agreement. To simplify the assessment of inter-annotator agreement, the scaling was reduced to a binary classification where 1 and 2 were marked as non-sarcastic and 3-5 sarcastic (recall that 3 indicates a hint of sarcasm and 5 indicates 'clearly sarcastic'). We checked the Fleiss' κ statistic to measure agreement between multiple annotators. The inter-annotator agreement statistic was $\kappa = 0.34$, which indicates a fair agreement (Landis and Koch 1977). Given the fuzzy nature of the task at hand, this κ value is certainly satisfactory. The agreement on the control set (anchor sentences) had $\kappa = 0.53$.

Results and Discussion

5-fold cross validation. Detailed results of the 5-fold cross validation of various components of the algorithm are summarized in Table 2. The SASI version that includes all components exhibits the best overall performances with 91.2% precision and with F-Score of 0.827. It is interesting to notice that although data enrichment brings SASI to the best performance in both precision and F-score, patterns+punctuations achieves comparable results with F-score of 0.812, with worse precision but a slightly better recall.

Accuracy is relatively high for all feature variations. The high accuracy is achieved due to the biased seed that contains more negative (non-sarcastic) examples than positive (sarcastic) examples. It reflects the fact that sentences that reflect no sarcasm at all are easier to classify correctly. The difference between correctly identifying the non-sarcastic sentences and the challenge of recognizing sarcastic sentences is reflected by the difference between the accuracy values and the values of other columns indicating precision, recall and F-score.

Surprisingly, punctuation marks serve as the weakest predictors, in contrast to Teppermann et al. (2006). These differences can be explained in several ways. It is possible that the use of sarcasm in spoken dialogue is very different from the use of sarcasm in written texts. It is also possible that the use of sarcasm in product reviews and/or in online communities is very different than the use of sarcasm in a private conversation. We also note that Teppermann et al. (2006) deal only with the sarcastic uses of 'yeah right!' which might not be typical.

Newly introduced sentences. In the second experiment we evaluated SASI based on a gold standard annotation created by 15 annotators. Table 3 presents the results of our algorithm as well results of the heuristic baseline that makes use of meta-data, designed to capture the gap between an explicit negative sentiment (reflected by the review's star rating) and explicit positive sentiment words used in the review. Precision of SASI is 0.766, a significant improvement over the baseline with precision of 0.5.

The F-score shows an even more impressive improvement as the baseline shows decent precision but a very limited recall since it is incapable of recognizing subtle sarcastic sentences. These results fit the works of (Brown 1980; Gibbs and O'Brien 1991) claiming that many sarcastic utterances do not confirm with the popular definition of "saying or writing the opposite of what you mean". Table 3 also presents the false positive and false negative ratios. The low false negative ratio of the baseline confirms that while the naive definition of sarcasm cannot capture many types of sarcastic sentences, it is still a good definition for a certain type of sarcasm.

	Precision	**Recall**	**Accuracy**	**F Score**
punctuatoin	0.256	0.312	0.821	0.281
patterns	0.743	0.788	0.943	0.765
pat+punct	0.868	0.763	0.945	0.812
enrich punct	0.4	0.390	0.832	0.395
enrich pat	0.762	0.777	0.937	0.769
all: SASI	**0.912**	0.756	**0.947**	**0.827**

Table 2: 5-fold cross validation results using various feature types. *punctuation*: punctuation marks, *patterns*: patterns, *enrich*: after data enrichment, *enrich punct*: data enrichment based on punctuation only, *enrich pat*: data enrichment based on patterns only, SASI: all features combined.

	Precision	**Recall**	**False Pos**	**False Neg**	**F Score**
Star-sentiment	0.5	0.16	0.05	0.44	0.242
SASI	**0.766**	**0.813**	0.11	**0.12**	**0.788**

Table 3: Evaluation on the evaluation set obtained by averaging on 3 human annotations per sentence.

Weight of various patterns and features. We present here a deeper look on some examples. A classic example of a sarcastic comment is:
"Silly me, the Kindle and the Sony eBook can't read these protected formats. Great!". Some of the patterns it contains are:

- me , the CW and *[product]* can't[7]
- *[product]* can't CW these CW CW. great !
- can't CW these CW CW.
- these CW CW. great!

We note that although there is no hard-coded treatment of sentiment words that are typically used for sarcasm ('yay!', 'great!'), these are represented as part of a pattern. This learned representation allows the algorithm to distinguish between a genuinely positive sentiment and a sarcastic use of a positive sentiment word.

Analyzing the feature set according to the results (see Table 2), we find that while punctuation marks are the weakest predictors, three dots combined with other features create a very strong predictor. For example, the use of 'I guess' with

[7] This sentence is extracted from a Sony eBook review hence only the phrase 'Sony eBook' is replaced by the [product] tag, while the 'Kindle' serves as a content word.

three dots:
"i guess i don't think very brilliantly.... well... it was ok... but not good to read just for fun.. cuz it's not fun..."

A number of sentences that were classified as sarcastic present excessive use of capital letters, i.e.:
"Well you know what happened. ALMOST NOTHING HAPPENED!!!" (on a book), and *"THIS ISN'T BAD CUSTOMER SERVICE IT'S ZERO CUSTOMER SERVICE"*. These examples fit with the theoretical framework of sarcasm and irony (see the Related work section) as sarcasm, at its best, emerges from a subtle context, hence cues are needed to make it easier to the hearer to comprehend, especially with written text not accompanied by audio ('...' for pause or a wink, '!' and caps for exaggeration, pretence and echoing). Surprisingly, though, the weight of these cues is limited and they fail to achieve neither high precision nor high recall. This can be attributed to the fact that the number of optional written cues is limited comparing to the number and flexibility of vocal cues, therefore written cues are ambiguous as they also serve to signify other types of speech acts such as anger and disappointment (sometimes manifested by sarcastic writing), along with other emotions such as surprise, excitement etc.

Context and pattern boundaries. SASI fails to distinguish between the following two sentences:
"This book was really good until page 2!" and *"This book was really good until page 430!"*.

While the first is clearly sarcastic (no context needed), the second simply conveys that the ending of the book is disappointing. Without further context, both sentences are represented by similar feature vectors. However, context is captured in an indirect way since patterns can cross sentence boundaries[8]. Imagine the following example (not found in the data set):

"This book was really good until page 2! **what** *an achievement!"*

The extra word 'what' produces more patterns which, in turn, serve as features in the feature vector representing this utterance. These extra patterns/features indirectly hint at the context of a sentence. SASI thus, uses context implicitly to correctly classify sentences.

Finally, here are two complex examples identified by the algorithm:

"If you are under the age of 13 or have nostalgia for the days when a good mystery required minimal brain effort then this Code's for you"
"I feel like I put the money through the paper shredder I shelled out for these."

Motivation for using sarcasm. A final insight gained from the results is a rather social one, maybe revealing an undertone of online social networks. As expected, there was a correlation between a low average star rating of a product and the number of sarcastic comments it attracted. This correlation reflects the psychological fact that sarcasm manifests a negative feeling. More interestingly, the products that gained the most sarcastic comments, disproportionately to the number of reviews, are Shure and Sony noise cancelation earphones, Dan Brown's Da Vinci Code and Amazon's Kindle e-reader (see Table 4). It seems that three factors are involved in motivating reviewers to use sarcasm: 1) the more popular (maybe through provocativeness) a product is, the more sarcastic comments it draws. 2) the simpler a product is the more sarcastic comments it gets if it fails to fill its single function (i.e. noise blocking/canceling earphones that fail to block the noise), and 3) the more expensive a product is it is likely to attract sarcastic comments (compare Table 4 with average star rating of 3.69 and average number of reviews of 1752 against 4.19 and 438[9] in the whole dataset (Table 1)).

Product	reviews	avg. star rating	price	sarcastic
Shure E2c	782	3.8	99$	51
da Vinci Code	3481	3.46	9.99$	79
Sony MDR-NC6	576	3.37	69.99$	34
The God Delusions	1022	3.91	27$	19
Kindle eReader	2900	3.9	489$	19

Table 4: Number of sarcastic comments vs. estimation of hype (number of reviews and average star rating) and price (amazon price at the date of submission).

We speculate that one of the strong motivations for the use of sarcasm in online communities is the attempt to "save" or "enlighten" the crowds and compensate for undeserved hype (undeserved according to the reviewer). Sarcasm, as an aggressive yet sophisticated form of speech act, is retrieved from the arsenal of special speech acts. This speculation is supported by our dataset but experiments on a larger scale are needed in order to learn how those factors are combined. We could summarize with a sentence from one of the reviews (unfortunately wrongly classified as sarcastic): *"It seems to evoke either a very positive response from readers or a very negative one."* (on the Da Vinci Code).

Conclusion

We presented SASI, a novel algorithm for recognition of sarcastic sentences in product reviews. We experimented with a large data set of 66000 reviews for various books and products. Evaluating pattern acquisition efficiency, we achieved 81% in a 5-fold cross validation on the annotated seed, proving the consistency of the pattern acquisition phase. SASI achieved precision of 77% and recall of 83.1% on an evaluation set containing newly discovered sarcastic sentences, where each sentence was annotated by three human readers.

We found some strong features that recognize sarcastic utterances, however, a combination of more subtle features served best in recognizing the various facets of sarcasm.

[8] Patterns should start and end with a high frequency word and punctuation marks are considered hight frequency.

[9] Average is computed after removing three Harry Potter books. Harry Potter books are outliers, each accumulated more than 5000 reviews which is highly uncharacteristic.

We hypothesize that one of the main reasons for using sarcasm in online communities and social networks is "enlightening" the mass that are "treading the wrong path". However, we leave this for future study.

Future work should also include incorporating a sarcasm recognition module in reviews summarization and ranking systems.

References

Brown, R. L. 1980. The pragmatics of verbal irony. In Shuy, R. W., and Snukal, A., eds., *Language use and the uses of language*. Georgetown University Press. 111–127.

Burfoot, C., and Baldwin, T. 2009. Automatic satire detection: Are you having a laugh? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 161–164. Suntec, Singapore: Association for Computational Linguistics.

Clark, H., and Gerrig, R. 1984. On the pretence theory of irony. *Journal of Experimental Psychology: General* 113:121–126.

Danescu-Niculescu-Mizil, C.; Kossinets, G.; Kleinberg, J.; and Lee, L. 2009. How opinions are received by online communities: A case study on amazon.com helpfulness votes.

Davidov, D., and Rappoport, A. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *COLING-ACL*.

Davidov, D., and Rappoport, A. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated sat analogy questions. In *ACL*.

Gibbs, R. W., and Colston, H. L., eds. 2007. *Irony in Language and Thought*. New York": Routledge (Taylor and Francis).

Gibbs, R. W., and O'Brien, J. E. 1991. Psychological aspects of irony understanding. *Journal of Pragmatics* 16:523–530.

Gibbs, R. 1986. On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General* 105:3–15.

Grice, H. P. 1975. Logic and conversation. In Cole, P., and Morgan, J. L., eds., *Syntax and semantics*, volume 3. New York: Academic Press.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. New York, NY, USA: ACM.

Katz, A. 2005. Discourse and social-cultural factors in understanding non literal language. In H., C., and A., K., eds., *Figurative language comprehension: Social and cultural influences*. Lawrence Erlbaum Associates. 183–208.

Kessler, S., and Nicolov, N. 2009. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *International AAAI Conference on Weblogs and Social Media*.

Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33:159–174.

Liu, J.; Cao, Y.; Lin, C.-Y.; Huang, Y.; and Zhou, M. 2007. Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 334–342.

Macmillan, E. D. 2007. *Macmillan English Dictionary*. Macmillan Education, 2 edition.

Mihalcea, R., and Pulman, S. G. 2007. Characterizing humour: An exploration of features in humorous texts. In *CICLing*, 337–347.

Mihalcea, R., and Strapparava, C. 2005. Making computers laugh: Investigations in automatic humor recognition. 531–538.

Muecke, D. 1982. *Irony and the ironic*. London, New York: Methuen.

Pang, B., and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, 271–278.

Pang, B., and Lee, L. 2008. *Opinion Mining and Sentiment Analysis*. Now Publishers Inc.

Polanyi, L., and Zaenen, A. 2006. Contextual valence shifters. In Shanahan, J. G.; Qu, Y., and Wiebe, J., eds., *Computing Attitude and Affect in Text*. Springer.

Popescu, A.-M., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 339–346. Morristown, NJ, USA: Association for Computational Linguistics.

Stingfellow, F. J. 1994. *The Meaning of Irony*. New York: State University of NY.

Tepperman, J.; Traum, D.; and Narayanan, S. 2006. Yeah right: Sarcasm recognition for spoken dialogue systems. In *InterSpeech ICSLP*.

Tsur, O., and Rappoport, A. 2009. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *International AAAI Conference on Weblogs and Social Media*.

Utsumi, A. 1996. A unified theory of irony and its computational formalization. In *COLING*, 962–967.

Utsumi, A. 2000. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics* 32(12):1777–1806.

Wiebe, J.; Wilson, T.; Bruce, R.; Bell, M.; and Martin, M. 2004. Learning subjective language. *Computational Linguistics* 30(3):277– 308.

Wilson, D., and Sperber, D. 1992. On verbal irony. *Lingua* 87:53–76.