Sarcasm Detection using Sentiment and Semantic Features

Prateek Nagwanshi and C. E. Veni Madhavan
Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

Keywords: Sarcasm Detection, Natural Language Processing, Opinion Mining, Computational Linguistics, Classification.

Abstract: Sarcasm is a figure of speech used to express a *strong* opinion in a *mild* manner. It is often used to convey the opposite sense of what is expressed. Automatic recognition of sarcasm is a complex task. Sarcasm detection is of importance in effective opinion mining. Most sarcasm detectors use lexical and pragmatic features for this purpose.

We incorporate statistical as well as *linguistic* features. Our approach considers the *semantic* and *flipping of sentiment* as main features. We use machine learning techniques for classification of *sarcastic* statements. We conduct experiments on different types of data sets, and compare our results with an existing approach in the literature. We also present human evaluation results. We propose to augment the present encouraging results by a new approach of integrating *linguistic* and *cognitive* aspects of text processing.

1 INTRODUCTION

Mining text, of both structured and unstructured nature, involves the use of a variety of models and techniques. The subtle applications of opinion, sentiment analysis require a good handling of human *figures-of-speech*. This challenging task involves inherently the recognition of a variety of *literary devices* such as metaphor, sarcasm, pun, wit. Our contention is that a combination of machine learning methods for large corpora analysis must be integrated with linguistic and cognitive processing algorithms.

In our evolving research programme termed DIAMETERS (Dialog, Metaphor, Expansion, Rewriting and Summarization) we seek a descriptive and prescriptive methodology. In this we capture the linguistic and cognitive aspects of human processing of textual utterances. The plan is to utilize the syntactic and grammatical information from *parse trees* and *pos tags* with semantic information from *typed dependencies* ([Stanford]) together with a proposed system of *cognitive tags*. The cognitive tags will typically consist of the *wh*-tags and the many-to-many relationships, $\mathcal{M}$, between the wh- question tags and the dependency types. For example, the *who*, *what* tags will be related to the types *nsubj*, *dobj* and a few others. At present we are building this set $\mathcal{M}$ of mappings. In our integrated system the standard grammatical parsing of a sentence S would be followed by traversals of the typed dependencies of S and the cognitive relationship maps $\mathcal{M}$ to elicit a natural understanding of S. Indeed, this stage will also involve the handling of appropriate ontologies pertaining to *world knowledge*. Another ongoing investigation is on the iterative, unification scheme for information on linguistic and cognitive parsing of two successive sentences. This stage will identify further contextual, discourse information from other devices of anaphora, named-entities. In this manner certain complex Natural Language Processing(NLP) tasks, that rely on *figures-of-speech*, such as metaphors, discourse, are expected to be handled *more* naturally and hence with *better* success for machine processing. A further possibility is that this approach will yield a language independent processing methodology.

In this work we address one such complex NLP task, namely *sarcasm detection*. At present we have not invoked the full power of the proposed methodology. We utilize certain amount of semantic information and proceed with a statistical classification methodology.

Sarcasm is a figure of speech that mostly conveys the opposite meaning to what is said. In verbal communication, the effect of *sarcasm* is brought out using voice tone, pitch, gestures and facial expressions. In written communication, such effects can not be used. However in text based communication sarcasm is used widely. It is used extensively in print media,

Nagwanshi P. and Veni Madhavan C..
Sarcasm Detection using Sentiment and Semantic Features.
DOI: 10.5220/0005153504180424
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2014), pages 418-424
ISBN: 978-989-758-048-2

social digital media channels such as e-mail, blogs, tweets etc.

A *sarcastic* statement is a witty or bitter remark that seems to admire someone or something but actually is actually used to insult or taunt. (e.g., *"I am trying to imagine you with a personality"*). A statement which contains sarcasm will generally depend upon some context. Hence it is very difficult to detect sarcasm in single sentence. In language and literary works, different kinds of sarcasm are used: self-deprecating, brooding, deadpan, polite, obnoxious, manic and raging. Humans can generally distinguish such subtle varieties of sarcasm. However, it is a challenge (González-Ibánez et al., 2011) to develop a computational scheme to even distinguish between *literal* and *sarcastic* statements.

Processing well-formed natural language sentences at *lexical*, *syntactic* and, to some extend *semantic* levels is an established science. However, handling figures-of-speech that have different properties has lagged behind because of the lack of remarkable computational theories and models.

People often use *sarcasm* and *irony* to express their opinions. There are many opinion mining tools. These fail to identify the *sarcastic* or *ironic* utterances. Usage of *sarcasm* is very common in web content like tweets, blogs and product reviews. Users express their feelings or reactions by using *sarcasm*, *irony* and other *linguistic* devices. To understand these opinions we have to go deep into the theory of sarcasm. When a writer wishes to say some negative remark about someone, he does not convey it directly, he uses *sarcasm* to say "a negative thing in positive words." (e.g., *"awww i love to get cute goodnight texts from no one"*). This example shows how people use *sarcasm* for conveying negative views. Words which exhibit politeness commonly used in sarcastic utterances. One of our aims is to capture the usage of positive words to convey negative things.

It is a challenge to automatically interpret and identify figurative usage of words. Our work is concentrates towards *sarcasm*. Sometimes it is difficult for humans to identify *sarcasm* using human intelligence, because it not so obvious. So semantic analysis may not be very useful. We model this by statistical models to predict the sarcastic utterances. Some of the sarcastic remarks came into picture because of usage. So we concentrate on statistical models and try to to develop a *supervised learning model* which can identify the *sarcastic* utterances.

2 RELATED WORK

Some major works in automatic processing of natural language texts for detection of sarcastic utterances are (Lakoff and Johnson, 2008; Utsumi, 2004; Tsur et al., 2010; Reyes et al., 2012; Riloff et al., ; González-Ibánez et al., 2011).

According to (Lakoff and Johnson, 2008) people often use *sarcasm* for insulting others. In *sarcastic* sentences, the speaker does not explicitly mention the negative interpretation of the sentence, so it is the responsibility of the listener to recognize speaker's intention.

(Utsumi, 2004) shows how linguistic style and contextual features plays a vital role in processing irony. He identifies irony on the basis of 3 types of patterns like: "Opposition", "Rhetorical question", and "Circumlocution."

Opposition is a statement in which the meaning is positive but is related to a negative situation, like:"This restaurant serves the dishes quickly."

Rhetorical questions are statements which contain a question as an obvious fact like: "Do you know the recipe for the dishes?"

Circumlocutions are statements weakly related to an expectation : "I think you are just going to buy the ingredients for the recipe." According to the author the degree of irony and *sarcasm* increases when the sentence is of type opposition, rhetorical question or circumlocution.

(Tsur et al., 2010) approached this problem by a semi supervised algorithm which has two stages: a pattern collection followed by a classification of sarcastic utterances. They conduct experiments on reviews of Amazon.com[1]. They use pattern matching and features based on punctuation to detect *sarcasm*. Each pattern is replaced by its general pattern like [product], [company]. Classification of a new review is based on the exact or partial match with stored patterns.

(González-Ibánez et al., 2011) have done a 3-way comparison of *sarcasm* with *positive* and *negative* sentiment carrying tweets. They use lexical and pragmatic features for the identification of *sarcasm* in Twitter data. Lexical feature is a combination of unigrams and dictionary based features. Pragmatic feature contains positive emoticons(smilies) and negative emoticons(frowning faces). According to them the auxiliary verb and the punctuation are also important features for identifying *sarcasm*. They conducted human evaluation for checking their algorithm and in

[1] www.amazon.in

the the end they conclude that neither the classifier nor the human judges perform well. Our approach is motivated by their work. They have not considered the sentiment based features.

(Reyes et al., 2012) focus on humor and irony processing. They compare humor and irony with different genres like politics, and technology. They use different features for the identification. These consist of ambiguity, polarity, unexpectedness and emotional scenarios. Ambiguity is a combination of structural, morphosyntactic and semantic layers. Structural ambiguity can be viewed as funny situations which occur most in the text containing humor.

(Liebrecht et al., 2013) tackle the problem by checking the presence of markers, intensifiers, exclamations on twitter data. They find that the markers like “lol” and “humor” and intensifiers like “awesome”, “lovely” and “fantastic” etc and positive exclamations like “yeah” , “yipee” and “wow” indicate sarcasm in tweets.

(Riloff et al.,) identify the sarcastic tweets that contain positive sentiment about an activity that is disliked normally. They determine that the pattern of positive sentiment expression followed by an activity, which normally people do not like to do, such as work or study will generally indicate sarcasm. The bootstrapping process takes a seed word and collects the negative situation phrases which are preceded by the seed word. For example, they collect a pattern like: *I love(positive sentiment word) being ignored(negative situation phrase)*. Then this procedure is applied in the opposite direction. They use these phrases as features for a SVM based classifier.

3 OUR ALGORITHM AND FEATURE SET

In this paper, we present a novel supervised learning algorithm for sarcasm identification. The algorithm has two modules: (i) Feature-extraction, and (ii) classification.

In the first step we extract all features described in the next section. Then we perform a SVM based classification. We evaluated our system on two data-sets: Twitter hash-tag data set and Quotation data set.

3.1 Feature Set

Our feature set covers the aspects: lexical, syntactic, semantic, pragmatic, politeness, sentiment flipping aspect.

- **Lexical feature** is based on n-grams(upto bigram) which occur more than twice in the training data. We build a dictionary from these words, then use it in a bag-of words approach.
- **Syntactic feature** is the combination of part-of-speech tags. We found by our studies that adverbs are used often in sarcastic utterances. Presence of adjectives is also a discriminative feature. We use certain n-gram combinations of part-of-speech tags which are very common in sarcastic utterances. We use the Stanford Pos-tagger (Toutanova and Manning, 2000) for generating the part-of-speech tags.
- **Semantic feature** determines whether there exists words which contradict or are nearly opposite of each other.
  For example:“I love being ignored.”. In the example two nearly opposite words are used which makes the utterance sarcastic.
  We use the WordNet (Miller, 1995) for finding the contradictory words.
- **Pragmatic feature** determines when the sentiment of the sentence differs from the emoticons or similes.
  For example: “I just love to wake up early in the morning” followed by frowning face simile. In the example the sentiment of the sentence is supposed to be positive but the use of the frowning face simile makes it sarcastic.
- **Politeness_rating** will determine if there are words like “extremely”, “too” used before any positive sentiment carrying word. This feature is very useful in discrimination because when we sarcastically want to express something then we say : “ You are extremely good”, instead of statement like “you are very good”.
- **Flipping of sentiment** will take care of the sentiment dissimilarity in the sentiment progression within a sentence.
  For example:“I love when I am sick and not even able to sleep.“ . In this example the sentiment of both clauses differs.
  We use Senti-WordNet (Esuli and Sebastiani, 2006) for checking the sentiment values of a word.

Our feature vector is of dimension 640. The breakup of feature set is as follows. lexical feature(1-610), syntactic feature(611-625), semantic feature(626-627), flipping of sentiment(628-635), pragmatic feature(636-638) and politeness_rating(639-640). The lexical feature vector (dimension 610) is an indicator vector based on a set of distinct content words collected from the twitter data. At present we use only unigrams, for match with the fixed set of most commonly occurring unigram. We could use bigrams and trigram phrases. This would expand the feature vector size. We plan to study the efficacy of using a more compact representation of the feature vector. SentiWordNet data is used to mark the positive and negative words in the sentence.

The classification accuracy was evaluated by employing the classifiers: Naive Bayes and Support vector machines(SVM). We show the results on the basis of some sample data.

4 EXPERIMENTAL SETUP

Although some work is reported in the literature to solve this problem, we do not have standard data set for evaluation of performance. i.e. used to evaluate the performance of the existing algorithms. So we are unable to find a gold standard data set for *sarcasm* identification. So we performed human evaluation of some data and used this as gold standard. We performed experiments on this data to check credibility of our algorithm.

Several experiments were performed to evaluate the capabilities of our algorithm. Based on the feature set discussed in the previous section, we develop the feature vectors corresponding to sentences. Then we use Weka (Hall et al., 2009)which is a tool for doing various Data Mining tasks like classification, clustering etc.

We conduct experiments on two kinds of data sets: Experiment-1 by using tweets as training data and Experiment-2 by using a quotation data set. Then we conduct human evaluation of the quotation data and try to verify the classification obtained by our algorithm.

Experiment-1

- **Twitter Data:**- A lot of opinion mining content is available in micro-blogging and social networking websites: like Twitter[2]. Twitter provides the facility to users to post and read the messages from others whom they follow. For most of the tweets the user provides a hash-tag. This is used to identify messages on a specific topic like sarcasm. It is used to group messages which contains similar topics. One can search the whole tweets by just typing the hash-tag and get the set of messages.

  In Experiment-1 the classifier-1 is trained with 200 instances (100 *Sarcastic* tweets and 100 *Positive* tweets). Classifier-2 is trained with 200 instances (100 *Sarcastic* tweets and 100 *Negative* tweets) A 10-fold cross validation method was used for the test set.

1. **Data Collection:**- Many of the tweets with the hash-tag, ”sarcasm“ are available in Twitter. We used the hash-tag data as a training set. We used the twitter streaming api[3] for collecting the tweets which are hash-tagged as #sarcasm, #positive, and #negative. The hash-tag is not a reliable source of classification, so we have manually checked some sample data.
2. **Preprocessing:**- We carry out an initial preprocessing of the data set. The preprocessing consists of following rules:
   - If a word starts with “http” ignore it.
   - If a word starts with “@” ignore it.
   - If a word starts with “hashtag” ignore it.
   - If a tweet is written in a language other than English, then ignore it.
   - The abbreviations and short acronyms are replaced in full.
   - The smilies are replaced by their indicative meaning word like “:)” with “happy-smilie” etc.
   - The emoticons are also replaced by their indicative meaning.
3. **Feature Extraction:**- Features are extracted from the training data according to our algorithm.
4. **Classification:**- This feature set is used for the classification purpose with the help of Weka.
5. **Results:**- We incorporate the features in a graded manner by including the different feature sets one after the other. First we consider only the unigram (lexical) features, then the lexical and syntactic features, then lexical, syntactic, semantic features, and finally we introduce the pragmatic (sentiment) features.

   The accuracy of the classifier after adding incrementally each feature is shown in Table 1. In Table 2 we give, the classification accuracy on the experiments between *sarcasm* vs *positive* (Sar-vs-Pos) and *sarcasm* vs *negative* (Sar-vs-Neg).

Experiment-2

1. **Standard Quotation Data:** Tweets are generally used in an unstructured format. So there is a need for cleaning up the data before processing the sentences. Also hash-tags are not fully reliable source to assume the class label. Hence there is a

[2] https://twitter.com/

[3] https://dev.twitter.com/docs/api/streaming

Table 1: Impact of features(Sar-vs-Pos): Twitter data.

Feature	Accuracy
Lexical	68.65
Lexical+Syntactic	71.3
Lexical + Syntactic + Semantic	73.4
Lexical + Syntactic + Semantic + Pragmatic	74.13

Table 2: *Sar-vs-Pos and Sar-vs-Neg: Twitter data.*

Classifier	Accuracy	Precision	Recall	F-measure
Sar-vs-Pos	74.13	0.742	0.741	0.741
Sar-vs-Neg	74.69	0.747	0.747	0.747

Table 3: Impact of features(Sar-vs-Pos): Quotation data.

Feature	Accuracy
Lexical	71.3%
Lexical+Syntactic	73%
Lexical + Syntactic + Semantic	75.4%
Lexical + Syntactic + Semantic + Pragmatic	76.02%

Table 4: *Sar-vs-Pos and Sar-vs-Neg: Quotation data.*

Classifier	Accuracy	Precision	Recall	F-measure
Sar-vs-Pos	76.02	0.76	0.76	0.76
Sar-vs-Neg	77.77	0.778	0.778	0.777

need for examination of standard and structured text like standard quotations.

In Experiment-2 Classifier-1 is trained with 200 instances (100 *Sarcastic* quotes and 100 *Positive* quotes). The Classifier-2 is trained with 200 instances (100 *Sarcasm* and 100 *Negative* quotes) A 10-fold cross validation method was used for the test set.

2. **Data Collection:** Many standard quotes and sayings are available in the web. We collect some standard *sarcastic*, quotes and some *positive* and *negative* emotion statements from Internet sources.[4] [5] [6]
3. **Preprocessing:** The words are stored after lemmatization and stemming.
4. **Feature Extraction:** Features are extracted according to our algorithm. In standard quotations, in general, pragmatic markers are not used: so we ignore the pragmatic feature here.
5. **Classification:** Classification experiment is performed with the help of the tool Weka .
6. **Results:** The accuracy of the classifier after adding incrementally each feature is shown in Table 3. In Table 4 we give, the classification accuracy on the experiments between *sarcasm* vs *positive* (Sar-vs-Pos) and *sarcasm* vs *negative* (Sar-vs-Neg).

We make some observations on the data sets and experiments. Twitter data is easily available in large volumes. However this data, although labeled with a #sarcasm tag, needs to be filtered (as also noted by (González-Ibánez et al., 2011)). The quotation data is a pre-filtered collection. Hence the set of about 100 quotations used by us a reasonable representation of sarcastic content. We plan to investigate the robustness of our approach based on different data sets. This would also strengthen our claim on the obliviousness of our approach to the source of data.

A 10-fold cross validation is used without loss of generality. More parsimonious use of the data is possible. Indeed we would study this with different features based on cognitive aspects. The cascading progression of including the features is based on the following principle. The lexical and syntactic features are gathered in a straightforward manner during parsing. The semantic features require processing with WordNet synsets. This stage is more time consuming. Finally, we have only considered few elementary pragmatic features.

5 COMPARISON WITH EXISTING APPROACH IN LITERATURE

We compare our results with (González-Ibánez et al., 2011). We do not have access to their data set for the classification experiments. However, we performed experiments on a similar genre (Twitter data) as used in their experiments. Their model is able to get 67.83% in Sar-vs-Pos and 68.67% in Sar-vs-Neg. Our results (about 74% accuracy) are better than the results of (González-Ibánez et al., 2011).

6 SIGNIFICANCE OF THE FEATURES

We determined the significance of features from the results of the classification exercises. These features play a key role in distinguishing between *sarcastic* and *non-sarcastic* sentences. We describe these significant features with respect to the Quotation data set.

[4] www.searchquotes.com

[5] www.oocities.org/kristensquotes

[6] www.coolnsmart.com

1. **verb_verb:**- This feature is the combination of two syntactic entities: verb followed by verb. The probability of first word being an auxiliary verb is very high in our finding.
   (e.g.,*"If your life is all about screwing things and* getting(VBG) hammered(VBN)*, then congratulations, you are a fool*)".
   In the above example "getting hammered" has same syntactic constructs which we want in this feature.
2. **a_j_n_v:**- It is combination of 4 syntactic entities: adverb followed by adjective followed by noun followed by verb.
   (e.g., *"If u want to look thinner, hang around people fatter than you..*").
3. **j_n_a_v:**- It is combination of 4 syntactic entities: adjective followed by noun followed by adverb followed by verb.
   (e.g.,*"If you do not want a sarcastic answer, then do not ask a stupid question!"*).
4. **politeness_rating:**- This feature considers the words like: "too",or "absolutely" etc used before a positive sentiment word. (e.g.,*"You are so clever that sometimes you don't understand a single word of what you are saying*".
5. **pos_neg pairs:**- This takes care of the case when there are both positive and negative kinds of words used. (e.g.,*"I love being ignored"*). In this example "love" is positive sentiment showing word, where "ignored" is negative sentiment word.

In addition, We describe some of the features which occur with nearly same probabilities in the *sarcastic* sentences. They also play a vital role in classifying *sarcastic* utterances.

- **sar_sign:**- When the sentiment of the sentence differs from the emoticons, similes or words which shows emotion like "wow", "yippie" etc.
- **a_v:**- This feature is for the combination of adverb followed by verb.
- **aj:**- This feature is for the combination of adverb followed by adjective.
- **n_j_v:**- This feature is for the combination of noun followed by adjective followed by verb.

7 HUMAN EVALUATION

For checking the credibility of our algorithm we conducted human evaluation of the Quotation data. We took the sentences in which most of the human judges agreed. We acknowledge the help of volunteers(4 high school teachers and 3 postgraduate students) in providing their input on data. There is a reasonable agreement (i.e., at least 4 of the 8 respondents gave scores above 6 in a scale of 0 to 10) on more than 75 of the 100 sample quotes provided for human evaluation. We use this collection of 75 quotes as our gold standard for computer experiments.

We applied our algorithm on this data and found that our algorithm predicts 75% of these as *sarcastic*. This is a reasonable agreement considering, the conclusions of (González-Ibánez et al., 2011), that human participants differ in recognizing sarcasm.

8 CONCLUSIONS

Use of sarcasm and other figures of speech are very common in our daily life. However automatic processing is a big challenge. We have introduced a new technique for recognizing the *sarcastic* utterances on the basis of semantics and lengths of sentiment progressions. We show that distinguishing between *sarcastic* and a *negative* sentiment statement is feasible up to 75% accuracy. We propose a set of features that may not yet lead to the best distinguisher. We can improve the performance further based on the integration of *linguistic* and *cognitive* features discussed in the introduction section, in the course of our ongoing work. The framework we are developing will provide further semantic, cognitive features, such as *structural or semantic distance between words or phrases conveying opposition and incongruity*.

REFERENCES

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422.

González-Ibánez, R., Muresan, S., and Wacholder, N. (2011). Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 581–586. Association for Computational Linguistics.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Lakoff, G. and Johnson, M. (2008). *Metaphors we live by*. University of Chicago press.

Liebrecht, C., Kunneman, F., and van den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Reyes, A., Rosso, P., and Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. Sarcasm as contrast between a positive sentiment and negative situation.

Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

Tsur, O., Davidov, D., and Rappoport, A. (2010). Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

Utsumi, A. (2004). Stylistic and contextual effects in irony processing. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, pages 1369–1374.