Antonio Gomez Lopez
Tech Review – CS410 Text Information Systems

Introduction

The purpose of this review is to present the concepts of:

1) Sentiment Analysis
2) Sarcasm Detection and challenges
3) Use of feature extractions for analysis
4) Approaches used for sentiment classification
5) Reported results

At the end of this review I summarized the most important points as part of the conclusion.

Sentiment Analysis:

According to Kumar and Goel [1], sentiment analysis is the process in which emotions are classified through text: emotions can be neutral, negative or positive.

According to Wikipedia [2], sentiment analysis refers to "[…] the use of natural language processing, text analysis, computational linguistics and biometrics to systematically identify, extract, quantify and study affective states and subjective information". It is applied to understand consumer opinions in surveys, social media, customer service satisfaction reviews among other areas.

Sarcasm Detection:

According to Bharti et al. [3], sarcasm is a specific type of sentiment where people "express their negative feelings using positive or intensified positive words in the text. While speaking, people often use heavy tonal stress and certain gestural clues like rolling of the eyes, hand movement, etc. to reveal sarcastic (cues)".

Given the fact that these cues are greatly minimized on text data, it is a real challenge for an algorithm to systematically detect sarcasm in text (hence providing a more accurate picture of consumer sentiment vs. a world where algorithms take what consumers write about at face value).

At a core, the process used to determine the presence of sarcasm in a text follows the steps indicated in Fig. 1 (borrowed from Kumar and Goel [1]).

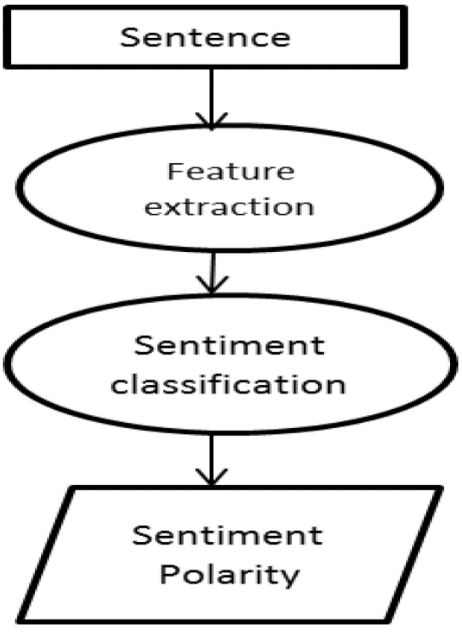

Fig 1. Process of Sentiment Analysis

Types of feature extraction for sentiment analysis

According to Bharti et al. [3] Sarcastic sentiment detection is divided into three different categories based on text features used for classification. Those are:

1) Lexical features: In this case, individual words in a text are “aggregated” using unigrams, bigrams, n-grams to try to identify sarcasm. Most recent work has combined this with syntactic features in order to improve the quality of the detector of sarcasm.

   Worth to note is a semi-supervised recognition processor by Davidov et al [4], where a semi-supervised model was developed in combination with an algorithm that scored texts using k-nearest neighbors.

   There has been countless research using unigram, bigrams, trigrams up to seven lexical features to detect sarcasm. Based on the results of this review there seems to be no-consensus on the ideal number n of “n-grams”

2) Pragmatic features: These relate to things such as emoticons (indicating positive ☺ or negative ☹ emotions), replies to a user or the use of “@” to refer to someone on Twitter. These are used frequently in platforms such a Twitter given the limitation in the number of characters.

3) Hyperbolic features: These include “punctuation and special symbols features”, as they are known in related literature. Those are comprehensive of symbols that are used to provide “intensity” to the text: Exclamation or interrogation symbols (including number of occurrences), the use of quotations, interjections (e.g. “wow”, “yay”), and use of capitalized text. Hyperbolic features also include the use of adjectives and adverbs to intensify the message. All of those can be useful in the task of identifying sarcasm in a text.

4) Contradicting features: According to Kumar and Goel [1], “[F]or sarcasm identification, ambiguity and coherent score is very helpful” (sic), those are represented as two binary features.

Sentiment analysis can be done using techniques combining the detection of these features in text. Take for instance the work of Bharti, Babu, et al. [5] where they test an approach that combines the use of lexical and hyperbolic features in order to improve the accuracy of their sarcasm detection model.

Algorithms used for classification in sarcasm detection:

At a high level, Kumar and Goel [1] provide the following approaches for algorithms used for sarcasm detection:

1) Rules based: Algorithms that use the presence of hashtags or emoticons embedded in a message in order to determine whether there is sarcasm.
2) Features based: Using some of the features described in the above section (including combinations of them).
3) Machine Learning based: Multiple approaches can be found using ML algorithms, including: SVM, Naïve Bayes, K-means, Gradient Boosting, Decision Trees, Random Boosting and Logistic Regression.
4) Deep Learning: This is a more recent trend following positive results that the use of DL has had in computer vision and pattern recognition. Deep Learning enables the automatic detection of features hidden in the data and claims to be superior to the above-mentioned ML techniques that require the “hand-crafted” creation of features.

Reported results:

Based on the sources I used to elaborate this review, I would say there is no consensus on which ML approach is best to use for the problem of sarcasm detection.

Ghosh and Veale [6] performed a comparison between traditional ML techniques (SVM in particular), with DL models (CNN, DNN, and combinations of them) to find out that in spite of respectable results using SVM (F-score > 0.66), DL models showed a clear superior performance (F-score > 0.84).

Consistent with the above and worth to mention is the work of Young, Hazarika, et al [7], where they review the use of Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Recursive Neural Networks, which are dethroning traditional ML techniques and becoming the state-of-the-art methods for sentiment analysis (and NLP) problems.

Conclusion:

Sarcasm is a form of speech that is often used (and misinterpreted) in online communities such as Twitter. Automatic recognition of sarcasm is difficult often because of the juxtaposition of ideas and sentiments, and the subtleties in the way these contrasts are performed.

In this report I presented a simple process to perform sentiment analysis and provided further explanation and examples coming from the work of different authors in the field of text processing and featuring extraction. Specifically, I presented four (4) different types of features: Lexical, Pragmatic, Hyperbolic and Contradicting.

The identification and classification of those features is deemed fundamental in the detection of sarcasm using rules-based and features-based classification methods. The latter makes use of traditional ML techniques (e.g. Naïve Bayes, SVM, Random Forest, etc.). Deep Learning methods allows for the automatic identification of features hidden in the text, and have been the subject of interest of NLP researchers in the recent past. Among them I mentioned the use of Convolutional Neural Networks, Recurrent Neural Networks and Recursive Neural Networks.

In terms of results, there is no clear consensus on what traditional ML method performs best among the many of them. However, based on the results referred on the papers consulted, it seems as if DL methods (or combinations of them) consistently outperform in the traditional metrics (F-Score, Accuracy, Recall) to traditional ML ones.

References:

[1] AI-Based Learning Techniques for Sarcasm Detection of Social Media Tweets: State-of-the-Art Survey, Kumar Yogesh, Goel Nikita.

[2] Sentiment Analysis, Wikipedia

[3] Sarcastic sentiment detection in tweets streamed in real time: a big data approach, Bharti S. K. et al.

[4] Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon, Davidov Dmitry et al.

[5] Parsing-based sarcasm sentiment recognition in Twitter data, Bharti S. K. et al.

[6] Fracking Sarcasm Using Neural Network, Ghosh A. and Veale T.