Sarcasm Suite: A Browser-based Engine for Sarcasm Detection and Generation

Aditya Joshi\textsuperscript{1,2,3} Diptesh Kanojia\textsuperscript{1,2,3}
Pushpak Bhattacharyya\textsuperscript{1} Mark Carman\textsuperscript{2}
\textsuperscript{1}Indian Institute of Technology Bombay, India
\textsuperscript{2}Monash University, Australia
\textsuperscript{3}IITB-Monash Research Academy, India
{adityaj, diptesh, pb}@cse.iitb.ac.in, mark.carman@monash.edu

Abstract

Sarcasm Suite is a browser-based engine that deploys five of our past papers in sarcasm detection and generation. The sarcasm detection modules use four kinds of incongruity: sentiment incongruity, semantic incongruity, historical context incongruity and conversational context incongruity. The sarcasm generation module is a chatbot that responds sarcastically to user input. With a visually appealing interface that indicates predictions using ‘faces’ of our co-authors from our past papers, Sarcasm Suite is our first demonstration of our work in computational sarcasm.

Introduction

Sarcasm detection gained attention from the sentiment analysis (SA) community for the challenges that sarcasm poses to typical SA systems. Several approaches to detect sarcasm have been reported Joshi, Bhattacharyya, and Carman (2016). This demonstration deploys five of our papers on sarcasm detection and generation. The demonstration titled ‘Sarcasm Suite’ is a browser-based engine that allows users to test these systems. At the time of writing this paper, Sarcasm Suite is available at: https://www.cfilt.iitb.ac.in/sarcasmSuite/.

Sarcasm Suite deploys four sarcasm detection modules and one sarcasm generation module. To the best of our knowledge, this is the first demonstration related to computational sarcasm. The demonstration will prove to be interesting as well as useful. We expect Sarcasm Suite to be pushed to its limits. However, errors made by Sarcasm Suite will provide ideas for future research in computational sarcasm.

Related Work

Sarcasm Suite is a deployment of the following past works by three out of four authors of this paper:

1. In Joshi, Sharma, and Bhattacharyya (2015), we use sentiment incongruity to detect sarcasm. We experiment with two sets of features: explicit incongruity features (which capture number of sentiment flips, sentiment subsequence lengths, etc.), and implicit incongruity features (which are phrases with implicit sentiment).

2. In Joshi et al. (2016b), we capture semantic incongruity via word embeddings in order to detect sarcasm. We experiment with two kinds of features: regular features and distance-weighted features.

3. In Khattri et al. (2015), we use an author’s historical context in the form of their twitter timeline to detect sarcasm in their tweets. This is a rule-based technique that calculates surface sentiment of a tweet and compares it with the author’s sentiment towards phrases in the tweet, in the past.

4. In Joshi et al. (2016a), we use conversational context using sequence labeling algorithms. The paper uses a dataset of transcripts from the TV show ‘Friends’.

5. In Joshi et al. (2015), we present a sarcasm generation module that responds sarcastically to user input. The module implements eight sarcasm generators, each covering a peculiar form of sarcasm.

Architecture

Figure 1 shows a schematic diagram of Sarcasm Suite. There are a total of five modules in Sarcasm Suite. A video of Sarcasm Suite is attached. This section describes different aspects of Sarcasm Suite.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
Sarcasm Suite is a unique engine that demonstrates five of our past works related to computational sarcasm: four sarcasm detection approaches and one sarcasm generation approach. This is our first demonstration of these papers. The engine offers a variety of approaches that use sentiment flips, word vectors, historical context based on twitter timelines, etc. as cues for sarcasm detection. Sarcasm Suite will enable the research community working in sentiment analysis to identify challenges encountered in sarcasm detection.

References


Joshi, A.; Tripathi, V.; Patel, K.; Bhattacharyya, P.; and Carman, M. 2016b. Are word embedding-based features for sarcasm detection? In EMNLP.


