Experimental Protocol – Sarcasm Detection

Adam Keppler
Department of Computer Science
Stanford University
akeppler@stanford.edu

Jennie Chen
Department of Computer Science
Stanford University
jenniechen@stanford.edu

Kais Kudrolli
Department of Electrical Engineering
Stanford University
kudrolli@stanford.edu

1 Hypothesis

Sarcasm is a form of figurative language in which the language used often contrasts with the intention of the speaker - for example, when very positive words are used to describe a negative situation. Sarcasm is inherently very context-dependent; a sarcastic statement taken out of context may appear perfectly sincere and vice versa. The task of sarcasm detection offers a unique challenge as even humans often have difficulty discerning whether a sample text exhibits sarcasm.

Previous approaches have chosen to manually extract sarcasm-related features or to rely on deep neural networks to learn sarcasm-related features but rarely both. One exception to this is the recent work of Son et al., in which features are learned from a bidirectional LSTM with attention and then combined with manually-extracted features in order to classify texts as sarcastic or not sarcastic [6].

Our goal is to improve on the work of Son et al.; while their work provided an effective method to integrate hand-crafted features and neural-network-derived features, we believe their work could be improved in three ways: correcting their preprocessing approach, adding context-related features that account for the characteristics of each user, and using attention to model the interactions between words in a sentence.

2 Data

Our project will utilize the SARC dataset as our corpus; introduced in 2017 by Khodak et al., it is currently the largest sarcasm corpus in use [4]. The researchers leveraged the plethora of user content on Reddit and the presence of a commonly-used sarcasm indicator "/s" to establish the dataset and manually verified a random subset of the dataset to provide a sense of its efficacy. Khodak et al. manually verified a random subset of the data and estimated a 1% false positive rate and a 2% false negative rate for the corpus. While both rates are low, the false negative rate is notable due to the sarcastic utterances only accounting for approximately 0.25% of the dataset; to help counter this issue, Khodak et al. provide both unbalanced and balanced datasets as the amount of noise is fairly small in a balanced setting.

The corpus is provided in two standard forms, a full collection and subset consisting of content just from the politics subset. The distribution of comments per dataset is shown in Table 1.

Khodak et al. also provide provide a benchmark for both basic baseline performance and human performance on the sarcasm detection task for the corpus. Due to the sheer size and breadth of the corpus and the importance of context (both of the text and of the topic) in detecting sarcasm, Khodak et al. measured baselines for both the entire corpus and a subset of data taken only from the r/politics subreddit to ensure that all their human evaluators had enough background information to comfortably detect sarcasm. Their best baseline, the Bag-of-Bigrams approach, achieved 75.8% accuracy on the full balanced dataset and 76.5% on the political

subset. Human performance, taken by majority vote of multiple reviewers, greatly surpassed their baseline with an accuracy of 92.0% on the full balanced dataset and 85.0% on the political subset.

	Training Set		Test Set	
	sarcastic	*non-sarcastic*	*sarcastic*	*non-sarcastic*
Main – balanced	128,541	128,541	32,333	32,333
Main – imbalanced	179,700	6,575,372	44,637	1,637,718
Pol – balanced	6,834	6,834	1,703	1,703
Pol – imbalanced	9,485	296,362	2,341	76,084

Table 1: *Distribution of sarcastic and non-sarcastic comments in each dataset.*

3 Metrics

We plan on using Macro F1-score as our primary metric. As sarcastic utterances are often present far less frequently than non-sarcastic utterances, it is best to use a metric that equally weights performance for each class. One of the advantages of SARC is that it provides both balanced and imbalanced datasets, allowing us to test performance of our models in either setting; this form of testing provides insight into generalizability, and we believe that by focusing on improving macro F1, we will be able to achieve the greatest level of generalizability.

We plan on using accuracy as our secondary metric for evaluation, as it is easily differentiable and can act as a useful proxy for our non-differentiable primary metric. Accuracy can be optimized by the cross-entropy loss function, allowing us to easily implement and train our models. It is also one of the most commonly reported metrics for our task. However, one drawback of accuracy is that it is less meaningful on imbalanced datasets. In our case, it may not appropriately judge how well the model correctly identifies comments that are actually sarcastic; thus, we keep it as a secondary metric, mainly for use with our balanced datasets.

It is important to note that as a part of our exploration of Macro F1, we will also be considering its subcomponents of per-class precision and recall in addition to accuracy and the full macro F1 value to help guide our decisions.

4 Models

We will be using the standard SARC Baseline as our reference baseline for standard performance. As stated above, the baseline of the Bag-of-Bigrams approach achieved an average F1-score of 75.8 on the full balanced dataset and an average F1-score of 76.5 on the political balanced subset [4]. On the political imbalanced subset, this approach achieved an average F1-score of 24.9. Virtually all work on SARC has continued to use the baseline values presented by Khodak et al. as the standard for performance on SARC.

While our goal is to explore the effectiveness of a hybridized model, we do not want to accidentally over look the importance of any one element. To ensure that we understand the impact of each component as well as its own potential, we willnot only test our hybrid model, but also a model based solely on each subcomponents. Our model can be subdivided into 3 main components: feature extraction, intra-attention, and deep neural models. We will run each of these components separately and then together in the final unified model. As we are running each element both as an independent model and as part of the hybrid, we will discuss each component uniquely in addition to the hybrid model.

4.1 Manually Extracted Features

We plan to manually extract both content-related and context-related features; these features will be used with a CNN classifier to classify comments as sarcastic or not sarcastic.

4.1.1 Content-related Features

We plan to use many different features derived solely from the content of each example. We will use various punctuation-related features, such as the count of question marks, periods, exclamation marks, and quotes as well as the number of capital letters and the presence of 3 or more consecutive vowels as indicators of

sarcasm, as they are often used to indicate exaggeration in text. In addition, we will look for the usage of interjections and uncommon words, which can also be an indicator of sarcastic text. Finally, we will use many sentiment-related features, including but not limited to the number of positive and negative words in the text as well as the number of incongruities in the text, which we define as a positive or negative word followed by a word of the opposite sentiment.

4.1.2 Context-related Features

To address the context of each comment, we plan to extract features related to the author of each example. Following the work of Bamman and Smith [1], we will build features related to each author's most common words and topics in their historical tweets. Since we do not have an easy way of scraping each author's comment history in the appropriate time period, we will consider all comments by an author in the main unbalanced corpus of the SARC dataset to be their comment history. For each author, we track the fraction of all their comments made in each subreddit as well as the fraction of sarcastic comments made in each subreddit. We also extract the most common words in the author's comment history (currently the 100 most common, although we will likely play around with this number) and then create binary indicators for each word's presence as a feature for the comment we are attempting to classify.

4.2 Intra-Attention

We utilize the multi-dimensional intra-attention introduced by Tay et al. [7] to provide insight into the relations between the words and highlight specific interactions with an utterance. The intra-attention network will produce a weighting for each word in the example, which will be used to produce a weighted sum of all the words in the sentence. We do this because often sarcasm is indicated by how certain words are used together.

The model will first look up word embeddings for each word in the sentence. Given l words in the sentence, the network will concatenate each word embedding with all other words in the sentence to create an $l \times l$ grid of word embeddings, representing every pair of words. Note that the pairs containing the same word from the sentence twice will be ignored. These concatenated representations will be run through a fully-connected layer, which produces another vector representation that can encode multiple word senses. This is followed by a ReLU, then another fully-connected layer to reduce each vector representation to a scalar.

Next, we take the max of each of the rows of the resulting grid. This operation picks out the most salient pair interaction for each word. We softmax the resulting vector, finally producing the attention weights. These are used to perform a weighted sum of all the input word embeddings. This final attention representation is passed to a final dense layer followed by a softmax operation to produce our final prediction.

4.3 LSTM Encoder

Tay et al. [7] and Kolchinski and Potts [5] show that LSTMs and GRUs are effective in encoding inputs for sarcasm detection. We will try a unidirectional LSTM encoder on its own since Tay et al. showed it worked well along side their intra-attention network. We may also experiment with a bidirectional LSTM, as used by both Kolchinski and Potts [5] and by Son et al. [6].

4.4 Hybrid/Unified Model

For our hybridized model, we will take the output vectors from our LSTM and intra-attention components and concatenate them with our feature vector. This combined vector will be passed into our CNN classifier to perform our final prediction. This closely follows the approach presented by Son et al., in which a network is hybridized with both automatically-extracted and manually-extracted features [6].

5 General Reasoning

Substantial progress has been made on the task of sarcasm detection through the use of deep models, including LSTMs, GRUs, and various attention networks, which are often able to identify complex patterns and features that are difficult to identify or extract. However, as noted by Ilic et al. these models generally use word-level representations, potentially missing important character-level cues such as use of punctuation or capitalization [3]. Ilic et al. address this by using a character-level representations of words, but we believe that by simply augmenting the features derived from the deep models with hand-extracted features, we can adequately account for these cues without drastically slowing down computation.

This is the approach taken by Son et al. as described previously. Although they worked with a different dataset (a roughly balanced corpus created by combining SemEval 2013 and 2015 datasets, as opposed to the SARC dataset we plan on using), their hybrid model showed over 6% improvement in accuracy compared to their best non-hybridized model [6]. Although not directly comparable, this lends to the hypothesis that such hybridization can be very effective, which we hope to test further on the standardized SARC dataset.

At the same time, we hope to make improvements to the work of Son et al. to address what we see as several shortcomings in their approach. For example, Son et al. choose to preprocess by stemming all words; however, we believe that this would remove important morphological information useful in detecting sarcasm. Additionally, Son et al. only use content-related features and do not account for the context of their examples, something that is very important when detecting sarcasm. To improve on their work, we plan to bring in author-related features such as common vocabulary and topics in an author's historical tweets, as done by Bamman and Smith [1]. Finally, while Son et al. use an attention mechanism to give more weight to words that are more important in a sentence, they do not properly capture interactions between words in a sentence, something that is very important in sarcasm. To this end, we hope to bring in the intra-attention mechanism defined by Tay et al. [7], which will help to capture contrast and incongruities in sentences that are often indicators of sarcasm.

Overall, our hypothesis follows closely to that of Son et al., who worked to create a model that combined deep learning with manually extracted auxiliary features. However, we believe that we can make several improvements to their approach, which we hope will lead to a significant increase in performance compared to Khodak et al.'s baseline when applied to the SARC dataset.

6 Summary of Progress

We have acquired our dataset and begun validation of its integrity. We have also implemented our data processor and feature extractors, though we may choose to add more features later. In addition, we have implemented the intra-attention network and its associated infrastructure as described by Tay et al.[7]. We are currently working on implementing a simple CNN and LSTM.

We plan to begin running and debugging each of our components in the next few days. Upon ensuring that each one can run separately, we will begin linking them together.

On the side, we are also investigating the disparity in our data. We've noticed that the distributions of our labelled examples do not match the numbers of either Kolchinksi and Potts [5] or Hazarika et al. [2]. Although this will not stop us from moving forward, we would like to figure out the cause of these differences. In addition, we've noticed that the number of comments available in the dataset's JSON file is very different from the number of comments used in the training and testing files. For example, for the politics subset of SARC, there are nearly 600,000 comments available in the JSON file, but only about 450,000 combined over all of the training and test CSVs, both balanced and imbalanced. We plan on contacting Khodak to ask about this discrepancy to determine if this is intentional.

References

[1] BAMMAN, D., AND SMITH, N. A. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media* (2015).

[2] HAZARIKA, D., PORIA, S., GORANTLA, S., CAMBRIA, E., ZIMMERMANN, R., AND MIHALCEA, R. CASCADE: contextual sarcasm detection in online discussion forums. *CoRR abs/1805.06413* (2018).

[3] ILIC, S., MARRESE-TAYLOR, E., BALAZS, J. A., AND MATSUO, Y. Deep contextualized word representations for detecting sarcasm and irony. *CoRR abs/1809.09795* (2018).

[4] KHODAK, M., SAUNSHI, N., AND VODRAHALLI, K. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579* (2017).

[5] KOLCHINSKI, Y. A., AND POTTS, C. Representing social media users for sarcasm detection. *arXiv preprint arXiv:1808.08470* (2018).

[6] SON, L. H., KUMAR, A., SANGWAN, S. R., ARORA, A., NAYYAR, A., AND ABDEL-BASSET, M. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access 7* (2019), 23319–23328.

[7] TAY, Y., TUAN, L. A., HUI, S. C., AND SU, J. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856* (2018).