Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230529

Detecting sarcastic expressions with deep neural networks

Zihang Huang

Shenzhen College of International Education, Shenzhen, China

s20481.huang@stu.scie.com.cn

Abstract. Following the ever increasing trend in social media such as Twitter, Facebook, and Instagram, automatic analysis of people's conversations and languages have become a problem of great significance for businesses and governments in attempt to understand and analyze people's habits, thoughts, and patterns towards different subjects of interests. Within the field of natural language processing, sarcasm detection has always been a difficult challenge for sentiment analysis. Recent years, there has been great interests shown by researchers towards sarcasm detection. Neural networks achieve huge success and advancements surrounding this topic, but reviews for this task are very limited and there's a lack of comprehensive review of the development of sarcasm detection so far. Thus, this paper aims to summarize and present the various methods directed towards sarcasm detection, the progress it has made, and examination of potential problems and availability for further improvements.

Keywords: sarcasm detection, deep learning.

1. Introduction

Sarcasm is a form of verbal irony with an intent specifically directed towards mocking or ridicule. It is very commonly found in micro-blogging and social network websites, and is used with the intention to intensify or dramatize the emotion in the message, as well as adding comedic sense to the text. Sarcasm refers to the uses of positive words or positive surface sentiments to describe negative or undesirable situation. For the sentence, "What a wonderful feeling it is, to buy a new phone and break it on the very same day.", sarcasm is indicated with the contrast of a positive sentiment "wonder feeling"and a negative sentiment of buying a phone and breaking it [1]. This contrasting sentiment between the words used and the actual message being conveyed can be hard to identify even by humans, let alone computers. In addition to the sentiment of words and subject communicated, accurate detection of sarcasm has to take into consideration the context of the whole message and context that each word appears to be in. For instance, when someone did something bad and another person replied "Well done!". Sarcasm can't be detected simply through the phrase "Well done!" as there are no contrasts in sentiment. Moreover, the uses of slang, dialects, punctuation, and symbols adds more complexity to this problem. These traits of sarcasm makes it a difficult task in sentiment analysis. Sentiment analysis has numerous implications in several areas such as business, health, and politics. For example, aiding business's analysis on customers' opinions and preferences on different goods and services. Or helping to investigate the affects of various events, rules, and regulations on civilians or specific groups of people. Accurate interpretation of sentence sentiments is crucial for analyzing and examining data generated by people on social media or microblogging websites [2].

DOI: 10.54254/2755-2721/5/20230529

However, the existence of sarcasm can confuse detection by interpreting its literal meaning rather than the actual intended message. This can result in evaluations that are the complete opposite of what is intended, driving false decisions and harming all the parties involved.

Initially, sarcasm detection relies on identifying the polarity of the verbs being used through a large emotion labeled corpus, or uses of lexical features including capital letters or repetitive usage of certain punctuation. Using handcrafted features, machine learning methods such as regression can be applied to classify a text as sarcastic or non-sarcastic. The first problem with this approach is its reliance on the background knowledge of the person who constructed the features, resulting in variation of features' quality as well as added uncertainty. Secondly, these features are generally extracted through statistical methods and only considers the surface sentiment of the words, and thus it can't effectively portray the deeper contextual meaning of words and sentences. Recent psychological studies indicates a strong relation between sentiment features and sarcasm, providing another method for this task. However, only relying on these characteristics are not effective as sarcasm can exist when there's no usage of the affective features described above. Deep neural networks are the current state-of-the-art in numerous natural language processing tasks like question answering, text summarization, and machine translation [3]. This approach has the ability to automatically learn and interpret necessary semantic features without the need of handcrafted ones through multi-layered non-linear transformation. This sparked attempts utilizing language models such as Word2vec and GloVe, creating models independent of feature engineering and handcrafted labeled corpus, but are insufficient when faced with affective tasks. More recent neural network models like ELMo and BERT are developed to overcome these problems through considering a word's contexts and its significance with self-attention mechanism.

Implementations of deep neural networks in natural language processing are rather successful in recent years and have overcome substantial amount of problems regarding sarcasm detection, yielding much better results and more stable classifications compared to methods used before. Nevertheless, there are very insufficient number of reviews on sarcasm detection, making it difficult and perplexing for newcomers who intends to focus on these fields of problems. Consequently, newcomers are not able to have a comprehensive intuition on the latest progress and methods proposed. Making it both time consuming and inefficient having to understand current achievements and causing badly directed research of this field. Furthermore, reviews can act as a summary of current height achieved, to help support future research and development, clarifying where to start and what can be improved. Hence, this paper organizes and summarizes the development of sarcasm detection in recent years. First, the definition of the task is classified, then common datasets and evaluation metrics are listed and briefly introduced. Moreover, mainstream deep learning methods for the task are explained and evaluated, their benefits and drawbacks. Lastly, the paper introduced newest methods and challenges for sarcasm detection as well as areas and potentials for further research.

2. Background

2.1. Task Definition

Most common research on sarcasm detection is focused on classification of given text. The aim here is on predicting whether a single sentence is contains sarcasm. This type of problem generally focuses on extraction of local sentiment features from the inputted text and the inputted text only [4]. Deep learning methods are then applied to these local features to create binary classification of the text's sarcasm. Research on this problem can be implemented to sentiment analysis of people's posts. For instance, classification of political opinions of the crowd in which sarcasm detection is applied to interpret microblogs and social media posts where users have the privilege to openly discuss and share their opinions, forming an interconnected ecosystem scattered with useful insights. Companies also take advantage of this ecosystem through analyzing public opinions about certain topics and provide real-time customer assistance to aid effective marketing.

DOI: 10.54254/2755-2721/5/20230529

Recent research has also focused on sarcasm detection for sequence of texts or dialogues. Each individual passage in the sequence is interpreted with all other passage in the sequence to classify sarcasm under additional contexts. This form of sarcasm detection has great significance in conversation or long passage processing. For instance, uses in chat-bots to create more intelligent, near-human interpretation of users' inputted texts, providing more accurate comprehension of the actual intended meaning of the text rather than literal understanding [5]. Improved chat-bots can be utilized in automatic question and answering, customer services, creating controversies and public discussions, and intelligent UI for any product such as cars or phones.

2.2. Datasets

Twitter: has a 280 character limit for tweets, so there are lots of abbreviations used by authors to fit the messages they want to send in one tweet. This along with colloquial languages adds many noise to the data. Despite these noise [6]. Twitter is still a very popular platform for collection of datasets to train and test sarcasm detection models due to the availability of Twitter API and its huge user base. Sarcastic labels of twitter datasets are collected through manual annotation and sarcastic hash-tags which are added by the author to sarcastic tweets they post. For automatic collection of sarcastic tweets, twitter API allows the user to search for the hashtags on the tweet. Generally, tweets with "#sarcastic" is collected and labeled as sarcastic tweets while tweets without the hash-tag is collected as non-sarcastic. However, non-sarcastic tweets can still be added with a sarcastic hash-tags while sarcastic tweets can exist with no hash-tags. This adds on to the noise of Twitter datasets.

News Headlines: Datasets from news headlines offsets the problems with Twitter datasets. Sarcastic headlines are collected from TheOnion [7], a news website focusing on reporting sarcastic versions of events. Non-sarcastic headlines are collected from HuffPost. These datasets collected from news headlines are more formal and are written by professionals compared to tweets on twitter where anyone can post using any language. TheOnion only publishes sarcastic news, this assures the sarcastic headlines collected from TheOnion has much less noise in comparison with twitter datasets. Furthermore, news headlines are self-contained, meaning it will not be semantically related to other headlines. Contrarily, tweets might be a reply to another tweet which would need additional consideration for context, adding uncertainty to extraction of sarcastic elements.

Internet Argument Corpus (IAC): IAC are public corporal constructed from forums and conversations surrounding political and social topics. IAC provides additional context for each post through linking posts to the post before them in a conversation, assisting extraction of contextual sentiments for the task. Machine learning using crowdsourced annotation will check whether each post is sarcastic. This generated a corpus of 4692 posts labeled with sarcastic or non-sarcastic.

Self-Annotated Reddit Corpus (SARC): SARC is a set of Reddit posts where sarcastic posts are indicated through adding "/s" at the end of the post and non-sarcastic posts are ones without "/s". Nonetheless, whether to add "/s" after a post is completely determined by the author which involves certain degree of noise in the dataset. For each post, details of its parent comments are also included, allowing the analysis of contexts involved.

2.3. Metrics

Precision: The number of correct positive predictions out of total positive predictions, calculated by the ratio of true positives to total positives predicted:

$$precison = \frac{TP}{(TP+FP)} \tag{1}$$

Recall: The number of correct positive predictions out of true positive cases, calculated by ratio of tree positives to total actual positives.

$$recall = \frac{TP}{(TP+FN)} \tag{2}$$

DOI: 10.54254/2755-2721/5/20230529

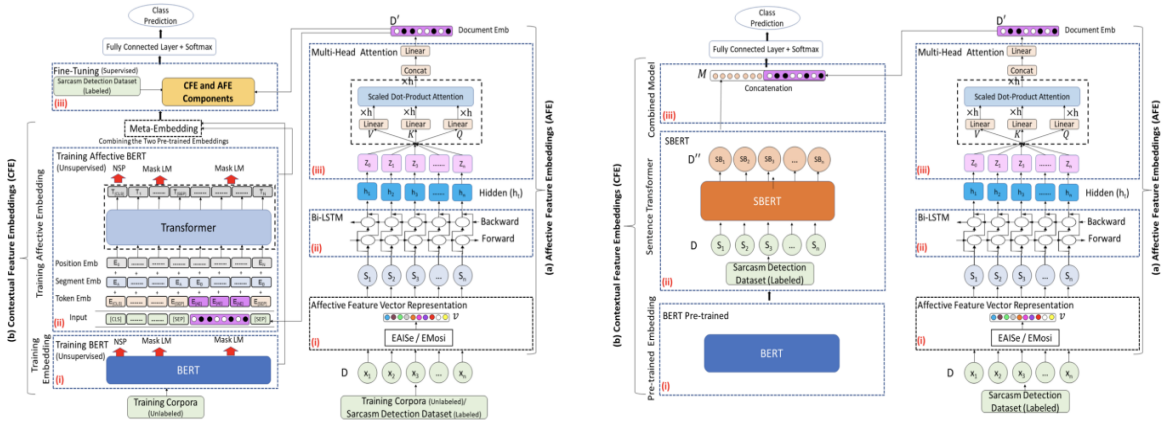

Figure 3. The architecture of the ACE 1 and the ACE 2.

Accuracy: The number of correct predictions out off all predictions, calculated by ratio of correct positive and negative predictions to all predictions.

$$accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{3}$$

3. Recent Researches

3.1. Literature 1

The existing BERT model can effectively approach tasks related to mood and feelings [8]. This model is further exploited in this paper to be directed towards sarcasm detection. Two models ACE 1 and ACE 2 are proposed to predict sarcasm given a text passage as input through Affective and Contextual Embeddings. Each model is made up of affective feature embedding, or AFE, and contextual feature embedding, or CFE, allowing the resulting representation to take into account the mood of sentences in the input.

In ACE 1, AFE and CEF are combined through training a BERT model while adding AFE into the input of BERT so embeddings for sarcasm detection can be obtained. For ACE 2, a fully connected layer using softmax combines AFE and CFE into a classifier trained from labeled sarcasm data.

AFE in ACE 1 and ACE 2 are the same with only minor differences in their input. To generate AFE, the input document is separated to sentences and either Emotion Affective Intensity with Sentiment Feature (EAISe) or Emotion Similarity Feature (EMoSi) is used to extract its affective features. Then, Bi-LSTM model captures information related to changes in emotions of the sentence in both forward and backward directions. A sequence of hidden state vectors ht, which is a concatenation of forward and back LSTM, is outputted to the next stage where the multi-head attention layer captures the significance of ht. CFE in ACE 1 is generated through training a BERT model on a collection of unlabeled texts over Masked Language Model and Next Sentence Prediction with the aim of minimizing combined loss function. Then, a new BERT model is trained by combining AFE to get the Affective BERT. Original BERT model, Affective BERT, and AFE are further combined and passed through fully-connected output layer that uses softmax to classify the input.

In ACE 2, pre-trained BERT contextual embeddings in feature-based approach is used to represent each input token from hidden layers of pre-trained model. SBERT is used to obtain a sentence embedding from an input sentence through triplet network structure. Then, a fully-connected layered with softmax is added on CFE and AFE components. For training and testing data sets, the paper created two corpus, Wiki and WikiSarc. Wiki contained a news corpus along with Wikipedia. WikiSarc contains Wiki, an Internet Movie Script Database, 60k tweets containing sarcasm hashtags, and 100k random tweets.

DOI: 10.54254/2755-2721/5/20230529

3.2. Literature 2

The paper suggested a Multi-Head self-Attention based Bidirectional Long Short-Term Memory (MHA-BiLSTM) network which also includes various handcrafted features [9]. This model aims to assist the identification of crucial parts of the input sentence to improve sarcasm detection's performance.

The MHA-BiLSTM consists of the following main parts: Word Embedding Layer - For every word xi in an input S, it is converted into its vector representation wi through an embedding matrix initialized by pre-training word embedding vector. Word Encoder Layer - The embeddings from the last part have the words independent of each other. To account for each word's context, bidirectional LSTM summarizes contextual information from both directions in the input through concatenating forward LSTM and backward LSTM to obtain hidden state representation ht for each word. Sentence Level Multi-Head Attention Layer - Multiple attention heads are used to give the corresponding importance that each part of the input plays in sarcasm detection through multiple factors. The hidden states are taken as input and multiplied with wk1 then pass init into tanh function. The result is then multiplied with wk2 and passed into softmax to normalize the importance of the weights. These weight factors are multiplied with all the hidden states of word to compute the sentence embedding M.

3.3. Literature 3

Multilinguality adds to the variety of user-generated content, adding to the difficulty of sarcasm detection [10]. In attempt to overcome this challenge, the softAtt BiLSTM - feature-rich CNN model is aimed to address sarcasm detection in mash-up language of Hindi and English.

The model proposed by the paper consists of three modules, the English language processing module, Hindi language processing module, and classifier module using convolutional neural network. For Hindi language feature extraction, all Hindi words are assigned with a tag by passing into Hindi POS tagger. Then, Hindi-SentiWordNet (H-SWN) determines a polarity score for each word. Language-independent n-grams with Tf-idf Vectorizer is used to construct HindiSenti Feature Vector by converting words into features. English processing module generates context vectors through BiLSTM with attention mechanism. The first layer in this component is embedding layer which maps words from English input into lower dimensional embeddings using pre-trained word embedding. Next layer is BiLSTM layer which outputs word features H from the concatenation of forward and backward LSTM to obtain the words' past and future context. The final layer is the attention layer. It uses differentiable and deterministic soft-attention mechanism to compute the weighted combination of all input states to find the significance that each word plays in the sentence and an attention score is given to each word. This attention score is used to calculate the weight for each word, that is then used to produce a hidden sentence feature vector through weighted sum function.

Pragmatic features also portray languages usage, so Pragmatic feature vector is created with 6 tuples. There's the frequency of recurring alphabetic character, number of exclamation marks, question marks, periods, uppercase letters, and single or double quotes.

3.4. Literature 4

Existing research have tried using deep learning to extract sentiment features and predict sarcasm through traditional machine-learning models. However, these work does not consider sentiment semantics. Because of this, the paper proposed a multi-level memory network based on sentiment semantics (MMNSS) to further exploit such information. Before inputting to the model, the input is embedded though an input encoding layer. Each word from the input is mapped to a word embedding to obtain a sentence matrix, it is concatenated with the position of each word to generate the final representation of the sentence S used as input for model.

First-level memory network is used to capture sentiment semantics used for sarcasm detection. SenticNet selects sentiment words as input for this part. LSTM encoder layer is applied to the sentiment input from last step to extract its features. The output matrix from LSTM and the query vector qw, which is the high-level representation, are inputted to memory network that uses attention

DOI: 10.54254/2755-2721/5/20230529

mechanism to extract sentiment semantics. The weight of each word is computed and multiplied with its corresponding embedding to get his. This product is passed through tanh function to yield a new representation h' is. The weight of this new representation is calculated according to the similarity between h' is and qw. Then softmax function normalizes the resulting weight matrix to obtain final weight matrix ais. The output vectors qsentiments are obtained through the sum of all products. Second-level memory network is used to capture contrast between sentiment and sentiment/situation. All words in the original input is applied with LSTM encoder to get fLSTM. This output is passed as input for the second-level memory network with qsentiments from last part as query obtain output vectors fmemory. Convolutional neural network with local-max pooling (LM-CNN) is used on sentence matrix S to capture local information which the memory network lacks to extract features fCNN. Finally, fCNN, fLSTM, and fmemory are concatenated and combined into a fully connected layer to feed to softmax layer for sarcasm detection.

4. Challenges

Annotation and labeling of sarcastic datasets has the problem concerning the quality of labels. Automatic hash-tag based labeling will inevitably be faced with the problem of noise. For the example of Twitter datasets, collection of non-sarcastic tweets through tweets without "#sarcastic" is not completely accurate as sarcastic tweets might not have that hash-tag. Finding non-sarcastic tweets through "#not sarcastic"generally will not reflect the form of normal conversation since "#not sarcastic" is mostly used when the topic or context is still somewhat related to sarcasm. Similar problem exists on hash-tag-based labeling of other datasets such as Reddit corpus. Manual annotation and labeling is time-consuming and has a concern for quality. Due to the subjectiveness of sarcasm, whether a text is classified as sarcastic is completely relied upon opinion of the annotator. Without a clear objective definition of sarcasm, uncertainty and noise of labels will also exist.

Sarcasm is not a frequent way to express sentiments in speech or conversation. Most of databases for model training and testing on sarcasm detection only has a small portion of the data being sarcastic. Data imbalance affects how accurate different performance metrics reflects a model's performance, so the choice of performance metrics used and degree of imbalance among datasets should be carefully considered for further research.

Existing research mainly analyzes contextual sentiments through sequence of text passages that links together. This is effective in a series of text based posts and replies, but there's only consideration for contextual sentiments within the local series of passages. Contexts not stated within the set of passages such as recent events, memes, and celebrities lacks sufficient analysis. In addition, non textual contexts, like images, although not a field of natural language processing, can play significant roles in sarcasm detection but currently lacks consideration.

5. Conclusion

Sentiment analysis has great potential to be exploited and manipulated. This is supported by the increasing trend of online social platforms, encouraging people around the globe to share and discuss thoughts and ideas on different topics. The large interconnected ecosystem of users all over the globe can provide beneficial information on fields such as politics, economics, and statistics. The complexity of sarcasm has always existed as a difficult challenge for sentiment analysis. This paper reviewed recent research on approaches to sarcasm detection through deep neural networks and evaluated their success and remaining problems. Current trend for sarcasm detection revolves heavily on better quality sentiment features and more sophisticated incorporation of context. As stated in challenges, noise of datasets and annotations adds uncertainty to feature embeddings for prediction, and contextual elements still has much to explore.

References

[1] Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., & Wright, B. (2020). Sarcasm detection using machine learning algorithms in Twitter: A systematic review. International Journal of

DOI: 10.54254/2755-2721/5/20230529

Market Research, 62(5), 578-598.

[2] Moores, B., & Mago, V. (2022). A Survey on Automated Sarcasm Detection on Twitter. arXiv preprint arXiv:2202.02516.

[3] Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., & Carman, M. (2016). Are word embedding-based features useful for sarcasm detection?. arXiv preprint arXiv:1610.00883.

[4] Mishra, A., Kanojia, D., Nagar, S., Dey, K., & Bhattacharyya, P. (2017). Harnessing cognitive features for sarcasm detection. arXiv preprint arXiv:1701.05574.

[5] Zhang, M., Zhang, Y., & Fu, G. (2016, December). Tweet sarcasm detection using deep neural network. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers (pp. 2449-2460).

[6] Babanejad, N., Davoudi, H., An, A., & Papagelis, M. (2020, December). Affective and contextual embedding for sarcasm detection. In Proceedings of the 28th international conference on computational linguistics (pp. 225-243).

[7] Jena, A. K., Sinha, A., & Agarwal, R. (2020, July). C-net: Contextual network for sarcasm detection. In Proceedings of the second workshop on figurative language processing (pp. 61-66).

[8] Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications, 32(23), 17309-17320.

[9] Lemmens, J., Burtenshaw, B., Lotfi, E., Markov, I., & Daelemans, W. (2020, July). Sarcasm detection using an ensemble approach. In proceedings of the second workshop on figurative language processing (pp. 264-269).

[10] Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., & Mihalcea, R. (2018). Cascade: Contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805.06413.

[11] M. H. Jafari, S. Samavi, S. M. R. Soroushmehr, H. Mohaghegh, N. Karimi, and K. Najarian, Set of descriptors for skin cancer diagnosis using non-dermoscopic color images,Sept 2016.