Sarcasm and Sentiment Final Paper

Akul Shah and Jeffrey Duan
University of Connecticut, akul.shah@uconn.edu, jeffrey.duan@uconn.edu

***Abstract* - This paper investigates the use of sentiment analysis in transformer models to detect sarcasm of tweets on Twitter. Sarcasm is a difficult language phenomenon to analyze, even for humans, as it involves saying the opposite of what is meant. This difficulty is further amplified in written communication, where there are no nonverbal cues to convey sarcasm. The paper explores the possibility of using transformer models to capture the true sentiment of irony and sarcasm by analyzing the relationship between words in a sentence. We utilize various transformer models, such as DistilBERT, RoBERTa, and our own transformer model made from scratch to analyze the sentiment of tweets, comparing the results to see if they can identify sarcasm and irony as close to humans as possible. This research aims to improve the accuracy of sentiment analysis on social media platforms and provide insights into people's thoughts and feelings about various topics. Finally explaining the limitations and challenges we faced when creating our model but how we believe the transformer models will eventually be able to mimic human accuracy in sentiment analysis of sarcasm.**

Index Terms – Irony, Model, Sarcasm, Sentiment Analysis, Transformers, Twitter.

INTRODUCTION

Social media platforms have transformed the way we communicate and express our opinions, being able to share our opinions with the world in a matter of seconds with Twitter being a prominent platform for users to share these thoughts and ideas. In most cases you can understand and distinguish the feelings behind these tweets. However, sarcasm and irony in conversation can be hard to distinguish, so adding in not being able to see the emotions through these social media platforms like twitter can make it increasingly difficult to accurately understand the true sentiment behind tweets. In recent years, sentiment analysis has emerged as a powerful tool taking advantage of recurrent neural networks and transformer models to identify the emotional tone of a message.

However we ask the question, if sentiment analysis can be used to identify sarcasm? This We were tasked to look into the benefits of sentiment analysis using transformer models to evaluate the given sentiment. Specifically targeting weather or not twitter statements are sarcastic or not, diving into the themes behind irony and sarcasm. We take advantage of various transformer models to analyze weather we can capture the true sentiment of Irony and Sarcasm.

WHAT IS SENTIMENT ANALYSIS

Sentiment analysis has become a significant part of natural language processing (NLP) which falls under the broader category of text classification, which is used to categorize text into different classes based on their content. The goal of sentiment analysis is to analyze and identify the underlying emotion of the text. The process involves analyzing the words, phrasing, and syntax used in the text to identify the specific emotions and opinions expressed. This is done using various machine learning algorithms and techniques that identify patterns and features in the text.

One of the key structures used in sentiment analysis recently is the transformer architecture, which is a deep learning technique that has shown significant improvements in natural language processing tasks. Transformer models are used to process the text input and analyze the relationships between words in a sentence expanding on recurrent neural networks, allowing for a faster and more accurate analysis of the sentiment expressed. With the increasing amount of data available on social media platforms, sentiment analysis using transformer models has become an important tool for businesses, researchers, and policymakers to understand public sentiment and make informed decisions. Sentiment analysis can be used for a range of applications, including social media monitoring, customer feedback analysis, and political analysis.

SARCASM

Sarcasm is a form of language that involves saying the opposite of what one intends to convey, often with the use of irony or satire. It is a common phenomenon in social media, and is inherently difficult to analyze, not just automatically but often for humans too. It has an important effect on sentiment, but is usually ignored in social media analysis, because it is considered too tricky to handle [2]. Thus, it is an intriguing concept not only because of its toughness for a computer but also because of how hard it is to tell for humans especially through written communication. In written communication we lack the nonverbal cues, such as tone of voice or facial expressions, that are present in face-to-face communication to tell if sarcasm is being conveyed. So seeing if not only a computer can pick up specific elements of what sarcasm is but see if there are any patterns to pick up sarcasm especially through that written text as stated above because of how hard it is. There have been some attempts and as stated above on how to do it using context clues, and expanding on some previous models made through hugging face we believe it will be a fun and interesting task.

TWITTER

Twitter is a popular social media platform that allows users to post and interact with short messages, known as tweets, that are limited to 280 characters. It has become a powerful tool for communication, news dissemination, and marketing, with millions of users worldwide. With Twitter's mass popularity one of the more intriguing things about twitter is because of its simplicity and mass use it is a

breeding ground for data. From people tweeting about politics, sports or tech, users sharing their feedback about a new shiny app, or passengers complaining to an airline about a canceled flight there are billions of thoughts being communicated daily [3]. Though with all the data provided from twitter it can be hard to make sense, understand how people are thinking and feeling about certain topics. So when we combine sentiment analysis with the mass amount of data on twitter it will allow us to make sense of all that data in real-time to uncover insights that could drive business decisions, and give us a broader understanding of people's thoughts.

Original Plan

Our original plan was to use the information found in the literature review by using the Hugging Face API to train and analyze a sentiment model comparing the results of the trained models. This would work first by gathering data using the Tweepy, you'll use Tweepy, an open source Python library to get tweets using the Twitter API [3]. Once we gather a set of tweets from Tweepy we would label them based on the sentiment of sarcasm or irony detected for our training. Then we were going to implement a hashtag tokenizer like GATE which uses a Viterbi-like algorithm to look for the best possible match that combines a set of known words that make up the hashtag, which would normally be characterized as a single token [2]. Then training them using both the DistilBERT, and RoBERTa, and comparing the results to see how the sentiment analysis performs to see if we could capture sentiment analysis that can identify sarcasm and irony as close to humans as possible.

Project Process

One of the first deviations we made is realizing how time consuming collecting labeling, and preprocessing the data would be so we sought out existing datasets to see if anything had any tweets grouped based on being ironic or sarcastic. We scoured a variety of datasets on sites like DataSource.ai, Numerai, and kaggle. In the end we were able to find a dataset on kaggle which was called Tweets with Sarcasm and Irony. The dataset had tweets classified into one of the 4 classes: Regular, Sarcasm, Figurative and Irony. The data had already been divided into a training and testing set so from there we imported the data (which were comma-separated values) and began to clean the data for training our models.

After we had found our data we began trying to implement the hashtag tokenization technique we had mentioned previously called GATE, while implementing that tokenization we realized the data we had gotten was not heavily reliant on using hashtags or looking at any true context clues which is what the GATE tokenization technique would rely on. So instead we devised a new strategy to incorporate building more of a transformer from scratch and the comparison of using the Hugging Face model, and seeing how each of the models would fare compared to each other and then comparing them to human research and numbers provided from our literature review. Although we did implement some elements of GATE tokenization when going through the preprocessing phases of data cleaning we did explicitly implement the GATE tokenization technique in its entirety as mentioned previously.

Following the data collection stages we began to preprocess the data for our first part of the project where we created a transformer model from scratch. The first thing we did was to create functions which would take the data rather than be labeled with the categories of regular, sarcasm, figurative and irony, we switched them to be labeled based on number where sarcasm and irony where labeled with a one and figurative and regular where labeled with a zero. Creating a new column with that information we removed the old column with the labels. Then we went through all the tweets, and removed any character and tried as much as we could to split and label any character, or sub words which were known (This is the preprocessing step which implemented certain ideas of GATE Tokenization. We also used can Named Entity Recognition (NER) to find the entity type of the tokens in the sentences, this will help us identify the type and relevance of numbers in our text data.

Finally for the first part we began building the model. First we split our data into training, testing and validation data. Then we began the tokenization step by splitting our tweets into the word and getting the vocabulary size of our data. Then we converted the sentences to token followed by the padded sequences in encoded format to create numeric encodings assigned to each word. We will only use the encoder block of the original transformers model (encoder-decoder) designed for this problem as the decoder is not needed. Following this we created our Multi- Headed attention model. Multi-head Attention is a module for attention mechanisms which runs through an attention mechanism several times in parallel. The independent attention outputs are then concatenated and linearly transformed into the expected dimension. Using all this information on the transformer model we created a and trained the attention model with 10 epochs on a batch size of 32 to see how that would compare.

While training that model we also used the Transformers library to train a model based on the distilbert-base-uncased pretrained model to compare to our transformer model from scratch. We began the process with a similar data preprocessing phase where we split the data into training and testing. We then had to relabel which of the elements looked to be sarcastic or ironic for the training. Then we cleaned and preprocessed the data prior to tokenizing, redefining any emoticons, and special characters that would be recognizable for the tokenizer. We also looked for URL links and reclassified those as well as any common contractions. Following this we had our new cleaned tweets with their respective class shown. After encoding the data using the transformers library we were ready to retrain the distilbert-base-uncased pretrained model, We trained the model using 5 epochs, and were ready to compare the results of the 2 models.

Results

The results from both models were very similar with our transformer model from scratch having an accuracy around 74% in identifying whether or not a tweet was sarcastic or ironic, in our testing and validation data. The model from scratch also had a loss of around 49.6% in the final epoch. Our model using the distilbert-base-uncased pretrained model had an overall accuracy of roughly 76% and a loss around 48%.

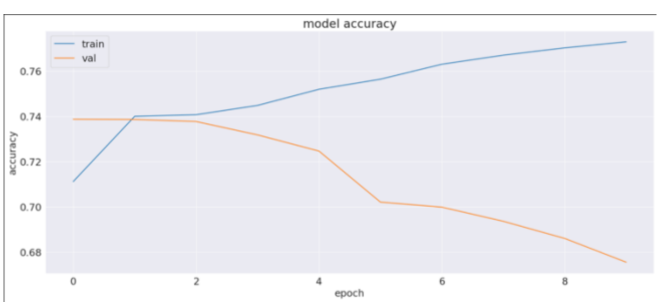

Figure 1: A Graphical analysis of the accuracy from the model we made from scratch.

When looking through our accuracies and how they compare to true human ability to detect sarcasm, and even current known models like the BMT-Net the models we created due follow a little further from the accuracies that those give. The broad multitask transformer network has shown to work very well in sentiment analysis cases with an accuracy of 94.0% which is the closest any model has come to in comparison to humans (97.8%) [1]. So it is fairly difficult to achieve such a high ability to detect sarcasm sentiment.

Limitations

In sarcastic text, people express their negative sentiments using positive words. This fact allows sarcasm to easily cheat sentiment analysis models unless they're specifically designed to take its possibility into account. This makes it so difficult to truly detect sarcasm and how the sentiment is being portrayed. One of the biggest challenges we faced was the base sentiment of understanding sarcastic sentiment is understanding the different context clues provided to the user. Those context clues we as humans have been trained to learn and understand but with machines they do not have as big of a luxury as understanding context, especially in textual form where the linguistic elements are non-existent making it less likely for someone to pick up those context clues.

In context we have four different types of sarcasm: propositional is where sarcasm appears to be a non-sentiment proposition but has an implicit sentiment involved. Embedded is when sarcasm has an embedded sentiment incongruity in the form of words and phrases themselves. Like-prefixed is sarcasm with a like-phrase which provides an implied denial of the argument being made. Illocutionary is a non-speech act (body language, gestures) contributing to the sarcasm [4]. Many of these however are hard to pick up as a machine and some are impossible to pick up through written forms such as text and that is what make sarcasm detection so tough. One of the most common forms of sarcasm we see in text is numerical sarcasm. Numerical sarcasm is related to changes in numerical values which then affect text polarity for instance someone saying: "It's 26 degrees outside and I am so cold." (Not Sarcastic) vs someone saying: "It's 98 degrees outside and I am so cold." (Sarcastic). But these also were very difficult to understand or predict without proper context of knowing what temperatures represent hot and cold etc.

Conclusion

Overall through the current expertise and advancements within sentiment analysis, the papers shown using transformer architecture models like a broad multitask transformer network have shown to work very well in sentiment analysis cases with an accuracy of 94.0% which is the closest any model has come to in comparison to humans (97.8%) [1]. However our models where unable to achieve those number close to human or even the best transformer model in relation to sarcasm as our model made from scratch had roughly a 74% accuracy, and using the pretrained distilbert-base-uncased had around 76% accuracy.

However, throughout this project we were able to get a deeper understanding of how sentiment analysis works and the methods and descriptions we would need to fully understand how the project went and everything that we were able to create and understand. The project itself has a lot of trials and tribulations in terms of what needed to happen, how everything was achieved and what we accomplished. Overall we were able to realize the changing aspects of sentiment analysis and how it could be used to detect sarcasm in the future. We believe that transformer models will be able to create a true human-like accuracy in achieving sentiment analysis focusing even further on the context clues and overcoming some of the challenges and limitations we had faced in the project.

This project we were able to fully create and understand transformer models and their capabilities with finding sentiment analysis, specifically looking at how impactful the sarcasm detection would be and the impact it could have on the future of machine learning and artificial intelligence in our world.

REFERENCES

[1] T. Zhang, X. Gong, and C. L. Chen, "BMT-net: Broad multitask transformer network for sentiment analysis," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6232–6243, Jul. 2022.

[2] D. Maynard and M. Greenwood, 'Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis', in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[3] F. Pascual, "Getting started with sentiment analysis on Twitter," *Hugging Face – The AI community building the future.*, 07-Jul-2022. [Online]. Available: https://huggingface.co/blog/sentiment-analysistwitter#how-to-do-twitter-sentiment-analysis-with-code. [Accessed: 18-Mar-2023].

AUTHOR INFORMATION

Akul Shah, Student, Department of Computer Science & Engineering, University of Connecticut.

Jeffrey Duan, Student, Department of Computer Science & Engineering, University of Connecticut.