Natural Language Processing - Text Classification

Twitter Sentiment Analysis

Nika Nizharadze

Computer Science
San Diego State University

Levan Badzgaradze

Computer Science
San Diego State University

Viraf Machhi

Data Science
San Diego State University

Introduction

Verner Vinge, a famous American science fiction author and retired mathematics & computer science professor at San Diego State University, writes in his 1993 Nasa conference paper, “The coming technological singularity: How to survive in the post-human era”: “I believe that the creation of greater than human intelligence will occur during the next thirty years.”[6] In his paper, he further discusses the conceptions of artificial intelligence among members of his contemporary society and its possible impact on the future of mankind. Whether Vinge’s anticipation of the creation of superhuman intelligence was accurate or not, one thing is clear – what Vinge labels as “weak” artificial intelligence (AI) is entering our everyday life and the implications of this development in regard to human race seem to stay a sensitive, highly debated topic. The attitude towards artificial intelligence has never been more relevant in human history. Even though most of the techniques used in machine learning, the driving force behind AI, were invented in the 1970s and following years, AI has gained unprecedented momentum only in the past couple of decades. Artificial Intelligence has become a buzzword and large research investments are being made to advance this field. How does the American population feel about AI development? Have numerous AI-based technological innovations skewed the American population’s perception? Have people grown numb to the idea of machine learning-driven technology penetrating the human lifestyle? These are several of the questions that motivated us to use four different models to conduct sentiment analysis of roughly 40,000 randomly chosen Twitter posts created in the month of December 2019. The goal of the analysis is to determine how the American population currently feels about artificial intelligence. Taking into account that Twitter is one of the most popular platforms the American population uses to share ideas and opinions about numerous controversial topics ranging from politics to social issues, we believe that sentiment analysis of Twitter posts has the potential of producing accurate analysis reflective of the entire target population.

Data Collection

Two distinct datasets were used for different purposes during the analysis.

IMDB's dataset

An open-source IMDB's dataset containing more than 50 000 manually labeled sentiments was utilized to train and evaluate the machine learning models.

Twitter dataset

Roughly 40,000 tweets with a hashtag AI (#AI) were downloaded through Twitter API. Collecting tweets from the Twitter database was accomplished by using the Tweepy library in python. Roughly 40,000 tweets were selected since Twitter API limits the maximum number of tweets that can be downloaded, and the process of crawling tweets stops after this threshold is reached. The Twitter dataset was used for the main objective of this paper - analyzing the American population's attitudes towards AI.

Data Pre-processing

The collected tweets were unstructured and required further processing. In order to clean the obtained data, the following steps were performed:

- Tweets were converted to lowercase using .lower() function.
- Duplicate tweets were removed from the dataset.

- UserMentions, links, and numbers were removed.
- Regular expressions were used to remove hashtags, emoticons, and punctuations.
- Stop words were removed from each tweet. Stop words are commonly used words like ‘of’, ‘the, ‘for’, etc., which have no semantic meaning and do not contribute to the sentiment of the sentence.

Upon completion of the data pre-processing, the dataset consisted of 13,812 unique tweets.

Models

1 Basic Recurrent Neural Network (RNN) Model

The first model was implemented using basic RNN, which takes sequences of words as input. For each neuron, the current word, as well as hidden state from previous layers, were fed as input.

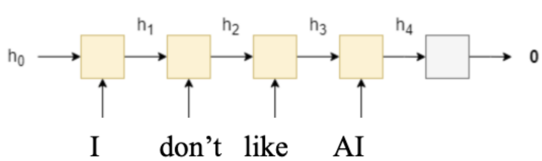

Figure 1

Figure 1 showcases how the tweet “I don't like AI” will be processed using the basic RNN model. Once the final hidden state is obtained, the data is fed into a sigmoid function to generate a 0 for negative sentiment or 1 for positive sentiment.

The initial parameters were set to zeros and one-hot vectors were used to create word embeddings. The challenge of this implementation was the sparse quality of one-hot vectors, which was resolved by converting sparse vectors into dense ones using the embedding layer. The dimension of one-hot vectors was equal to the vocabulary size, which is roughly 25000 words, the dimension of the embedding layer – 250, and the number of hidden layers – approximately 500. The output of either 0 or 1 is generated by the model.

2 Bidirectional LSTM

With the goal of improving the performance of the basic RNN-based model in mind, the second model was created as the Bidirectional LSTM. Instead of initializing word embedding to zeros, pre-trained vectors were used. Vector representations for words were obtained via using Global Vectors for Word Representation (GloVe) model, which performs training on word-word occurrences from a corpus and showcases linear substructures of the word vector space. Furthermore, the optimizer function was changed from SGD (Stochastic Gradient Descent) to Adam optimizer which is an adaptive learning rate method and uses the first two moments of the gradient to adapt the learning rate for each weight of the neural network.

The previous model (the basic RNN model) calculated the probability of sentiments using a current word and the hidden state from previous layers, however bidirectional LSTM models can preserve the information from the past as well as from the future.

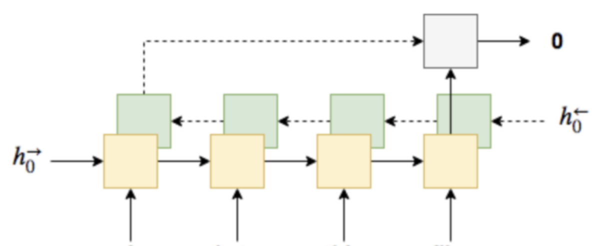

Figure 2

When applying this model to the Twitter dataset, it was observed that the model was overfitting the dataset. Through regularization technique dropout with a probability of 50%, the issue was successfully bypassed.

3 FastText Model

The basic RNN and Bidirectional LSTM modes were computationally heavy and required a powerful GPU for training. With this shortcoming in mind, a less computationally demanding FastText model (proposed by

Armand Joulin) was implemented. The FastText model calculates the n-grams and appends them at the end of the sentence. The embedding layer of the FastText model calculates the average of all word embeddings using avg_pool2d() function and then feeds this data into a fully connected layer.

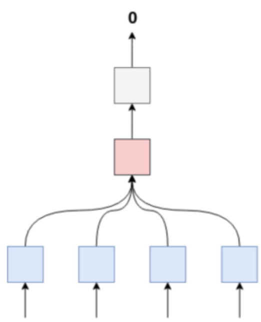

Figure 3

The FastText model uses bi-grams – is a pair of words that appear consecutively in the sentence.

4 Convolutional Sentiment Analysis

The traditional use of Convolutional neural networks (CNNs) is image recognition tasks. However, it can also be successfully applied to natural language processing problems such as text classification. Extracting the important features from the images is one of the widely known strengths of SNN and it is easy to see how the text classification problem could benefit from the same paradigm. However, while images are two (grayscale image) or three dimensional (2D image x RGB layer), any textual data is limited to only one dimension. For the successful implementation of the CNN model, the one-dimensional textual data were represented in two dimensions – each word was represented along one axis and elements of the vectors along another. Afterward, filters were created encompassing two words at a time; different filters being used to extract different features from the text. Next, the pooling layer was applied to the output of the convolutional layer. In contrast with an average pooling layer of the "FastText" model, the CCN model utilized max-pooling extracting the most important n-grams from the sentence.

The convolutional Sentiment Analysis model uses 100 filters with 3 distinct sizes resulting in 300 different n-grams modeling the most important features. As a final step, those n-grams were concatenated into a single vector and fed into a linear layer to predict the sentiment.

Techniques

PyTorch, the machine learning python library was utilized to implement all four models presented in this paper. Other software libraries used to produce our results included: Pandas, a software for building data frames from the dataset presented; GloVe model – for word embeddings; Matplotlib python plotting library along with online resources like rapidtables.com – for creating plots; Torchtext package – for acquiring an IMDB dataset; spaCy library – for tokenizing the data in preparation for the machine learning models; Twitter API and the python library Tweepy –for building the dataset from Twitter posts.
Considering the fact that the implemented models required powerful GPUs for training, Google Colab, a free cloud service providing GPUs, was successfully utilized for no extra cost.

Results and Analysis

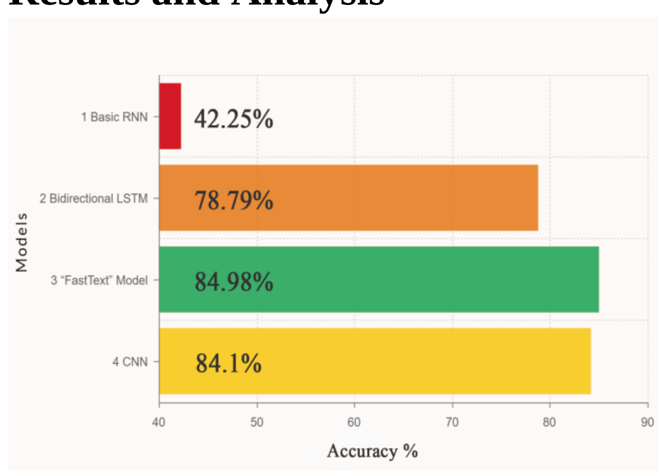

Figure 4

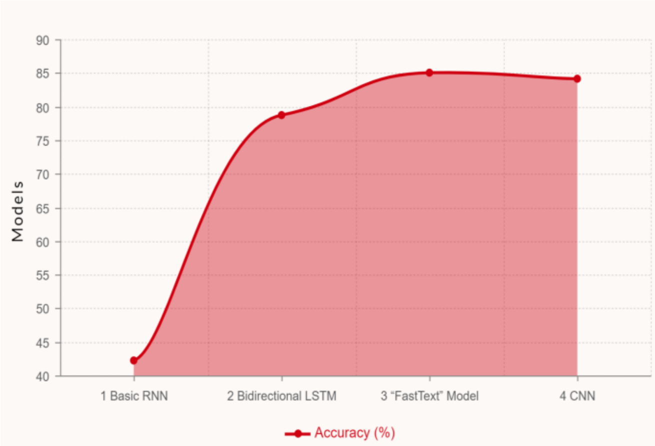

Figure 5

The graphs Figure 4 and Figure 5 show the corresponding accuracy for each model implemented on the testing set. Poor performance was expected from the RNN model considering its inability to capture long term m dependencies in the sentences. This challenge was overcome by the Bidirectional LSTM model almost doubling accuracy. It was unexpected to see the CNN model perform better than bidirectional LSTM judging from the fact that CNN models specialize in image processing. the FastText model and the CNN model demonstrated similar performances, the FastText showing the best result of 84.98% accuracy.

The FastText model was selected as the best candidate for labeling the tweets about AI stored in the dataset. Negative tweets were labeled as 0's and positive ones were represented with a value of 1. Manually inspecting the validity of sentiments labeled by the FastText model in the dataset of roughly 13,812 tweets is extremely labor-intensive. Instead, Figure 7 and Figure 8 demonstrating the most frequently used words for positively and negatively labeled tweets were created in a WordCloud representation.

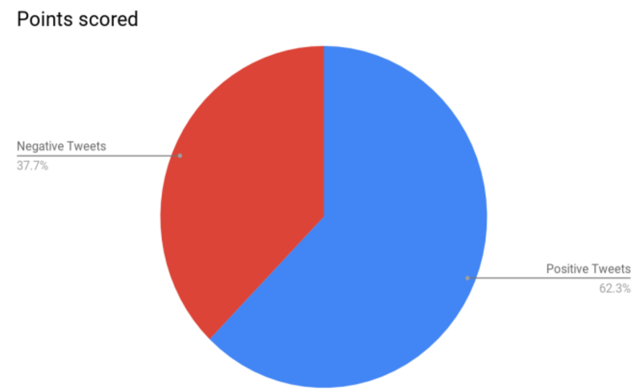

Figure 6 (Blue – positive, Red - negative)

Out of 13812 tweets about AI, 8605 (~62.3%) were positive and the remaining 5207 (~37.9%) were negative.

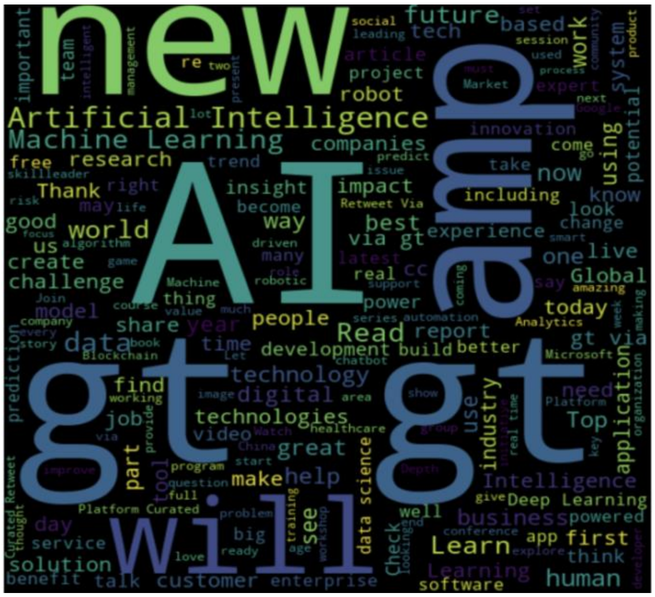

Figure 7

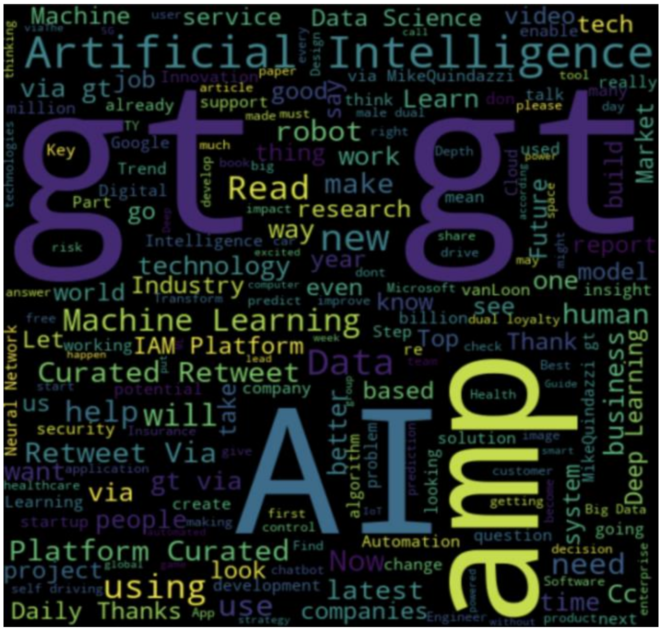

Figure 8

Figure 7 represents the WordCloud of the most frequently used words in positively labeled sentences, while the Figure 8 shows the WordCloud of the most repeated words in negatively labeled sentences. As expected, there is a big overlap between these two, however, there are few distinctions that are worth pointing out. Positive tweets contained words, like ‘new’ ‘research’, ‘impact’, ‘insight’, ‘will’, ‘development’, words usually used in positive syntax, so we can assume that the model labeled the dataset relatively well.

Conclusions

Four different models viable for natural language processing tasks such as analyzing sentiments of sentences were discussed and contrasted in this paper. Moreover, the dataset for tweets about artificial intelligence was acquired and the best performing model – FastText was selected to label the sentiments of the acquired tweets. The majority of the posts about AI (roughly 62.3%) were labeled positive, which suggests that the majority of the target population have positive feelings about artificial intelligence. The somewhat even distribution of sentiment might indicate that the American population still has mixed feelings about AI and its implication. Lastly, the slight trend might be observed – skewing public opinion towards a positive attitude hypothetically explained by the highly beneficial nature of the abundant AI-driven technology in the modern-day USA.

Future work

For future work, we plan to enhance our models from binary sentiment classification to multi-class classification. Having a wider range of emotions to analyze can give us a deeper understanding of the data.
Besides, the model could be implemented into a commercially available software capable of targeting individuals' sentiments using a custom keyword and database to create an appropriate context.

References

1. Bagheri, et al. “Sentiment Analysis of Twitter Data.” *ArXiv.org*, 16 Dec. 2017, arxiv.org/abs/1711.10377.
2. Conneau, Alexis, et al. “Very Deep Convolutional Networks for Text Classification.” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Association for Computational Linguistics, 2017, pp. 1107–16. DOI.org (Crossref), doi:10.18653/v1/E17-1104.
3. Jiang, Wei, and Zhong Jin. “*Integrating Bidirectional LSTM with Inception for Text Classification.*” 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2017, pp. 870-75.

4. Joshi, et al. "Twitter Sentiment Analysis System." *ArXiv.org*, 20 July 2018, arxiv.org/abs/1807.07752.
5. Joulin, Armand, et al. "Bag of Tricks for Efficient Text Classification." *ArXiv:1607.01759 [Cs], Aug. 2016. arXiv.org,* http://arxiv.org/abs/1607.01759.
6. Vinge, V. (1993). The coming technological singularity: How to survive in the post-human era. *Science Fiction Criticism: An Anthology of Essential Writings*, 352-363.

Helpful Articles

Al-Masri, A. (2019, February 13). Creating The Twitter Sentiment Analysis Program in Python with Naive Bayes Classification. Retrieved from https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed.

Al-Masri, A. (2019, February 13). Creating The Twitter Sentiment Analysis Program in Pythowith Naive Bayes Classification. Retrieved from https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed.

Special thanks to @bentrevett (Ben Trevett) for providing high level descriptions for implemented models.