Detecting Hate Speech in Tweets Using an Attentive Neural Network

Samuel Piltch
Brooklyn Technical High School
samuelpiltch@gmail.com

1 Introduction

Over the last few years, social media has seen rapid growth in global use. This "social media revolution" has brought the world closer together and has accelerated the speed at which information is shared and the rate at which relationships form. Social media also has an impact on the world's events and culture, being strongly influential in people's lives and empowering those to take a stand (Benioff, 2012).

As the influence of social media increases, social media companies strive to create safe platforms for their users. However, a recent rise in hate speech postings has obstructed this goal and has become the most challenging problem social media companies struggle with. Hate speech is formally defined as any communication that disparages a person or a group on the basis of some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics (Nockleby, 2000).

Hate speech online has also contributed to the recent rise in hate crimes. Experts believe that as more people move online, racist, misogynistic, and homophobic individuals have found communities of similar users that reinforce their views and encourage them to carry out violent acts. For instance, the 2018 Pittsburgh synagogue shooter was a participant in the social media network Gab, whose soft rules have attracted extremists banned by larger platforms. In Germany, a relationship was found between anti-refugee Facebook posts by the far-right Alternative for Germany party and attacks on refugees such as arson and assault (Laub, 2019).

This paper applies an advanced attentive neural network on a set of labeled tweets to predict if a tweet is either hate speech, offensive speech, or neither.

2 History

History of social media

Social media has grown immensely in the last decade. It plays a large role in our lives as we use it to connect with friends and family, catch up on current events, and entertain ourselves. In 2019, a total of 2.77 billion social media users exist globally, but just a few years ago in 2010, there was only a total of 0.97 billion social media users. That's an impressive 285% user growth over a short 9 year period ("Number of social media users worldwide 2010-2021," n.d.).

The emergence of social media doesn't occur until the 1980s and 1990s, two decades after the invention of the first Internet prototype, when personal computers became more ubiquitous. The original social media sites were Six Degrees and Friendster, which no longer exist today. Six Degree was founded in 1997 and its number of users reached a peak of 3.5 million before shutting down in 2001. Friendster emerged in 2002 and its number of users grew to over one hundred million. They were followed by the launch of LinkedIn in 2002 and later MySpace in 2003. In 2004, Facebook was founded and in 2008 it replaced MySpace as the top visited site. Twitter was created in 2006, Instagram in 2010, and Snapchat in 2011 (Terrell, n.d.).

History of hate speech classifiers

As an attempt to remove hate speech from their social media platforms, social media companies are implementing machine learning models that can classify postings as either hateful or not.

Early approaches to hate speech detection trained on manually extracted features such as n-grams, part-of-speech tags, and lexicons to represent texts. This allowed for high precision and an understanding of what the model was doing under the hood. However, this approach was inefficient and soon after hate speech classification transitioned to a neural network approach (Wang, 2018).

The neural network models that social media companies employ learn from a set of human-labeled postings, however, they have many flaws and can be tricked with different workarounds. A group of researchers found that certain models could be broken by inserting typos, using leetspeak (i.e.: l33tsp34k), adding extra words, or inserting and removing spaces between words. The success of each technique varied depending on the algorithm, but each classifier was hindered by at least some of the methods (Wired, 2018).

History of attentive recurrent networks

The original formulation of a recurrent neural network was made by John Hopfield in 1982. He developed a very simple recurrent neural network that became the first attempt at understanding the process underlying associative memory. In 1997, Hochreiter and Schmidhuber invented Long Short Term Memory (LSTM) networks. However, only recently (in 2007) did LSTM models begin to revolutionize the field of machine learning as computers became powerful enough to train them. LSTMs first came to fame for outperforming all preceding models in certain speech applications and later gained popularity in recognizing handwriting. In 2015, Google's speech recognition model improved by 47% by using a LSTM network (Atanasov, 2018).

3 Mathematical Background

Natural Language Processing

Natural Language Processing (NLP) is a branch of machine learning that aids computers in understanding, interpreting, and manipulating human language. NLP tasks break down language expressions into shorter elements, attempt to develop relationships between these elements, and explore how these elements work together to develop an understanding of the overall meaning of an utterance. NLP is used in many common and helpful applications such as email spam filters and virtual assistants like Siri and Alexa ("What is Natural Language Processing?" n.d.).

Tokenization - The process of taking each word in a sentence and converting it into an integer value that corresponds to its word vector.

Vector Word-Embeddings - A vector that represents the meaning of a word as multidimensional floating point numbers where similar words are mapped to adjacent points in geometric space.

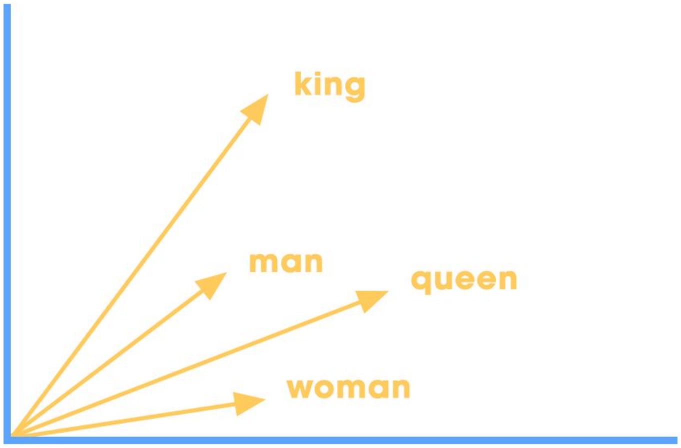

Figure 1. A visual representation of vector word embeddings. Words with a similar meaning are spatially near to each other. These vectors also represent relationships between the meaning of these words, for instance, in this example: king - man + woman = queen.

Vector Embedding Models - A model that generates vector word embeddings by learning from a large data set of sentences.

Attentive Recurrent Networks and LSTMs

Recurrent networks work differently from traditional feedforward neural networks as they take not only the current training example as input but also what the model has seen previously. The recent past examples act as a second input for the model and are stored in the recurrent network's hidden state. The process of carrying memory forward can be mathematically defined as:

$$h_t = \phi(Wx_t + Uh_{t-1})$$

The hidden state at time *t* is h_t. The input at time *t* x_t is multiplied by a weight matrix *W* and added to the hidden state of the previous time step h_{t-1} multiplied by its own transition matrix *U*. The weight matrices, *W* and *U*, act as filters that determine the importance of the present input and the previous hidden state, respectively. The weight matrices are adjusted using back propagation so that the model generates the lowest error when comparing the model's predicted output and the training point's output label. The ϕ symbol represents a squashing function which condenses the sum of the weighted input and past hidden state into a range of values between -1 and 1 or 0 and 1 depending on the chosen squashing function. Because this feedback loop occurs at each time step in the series, each hidden state contains traces of all the previous hidden state values for as long as memory can persist.

Recurrent neural networks suffer from short term memory, as they can overwrite their memory at each time step in a fairly uncontrolled fashion. Long short term memory (LSTM) cells solve this problem by transforming its memory in a precise manner. LSTMs have multiple learning mechanisms that alter the model's memory. By using specific learning mechanisms that decide which pieces of information to remember (the forget gate), which to update (the input gate), and which to pay attention to (the output gate), LSTMs can retain information over long periods of time.

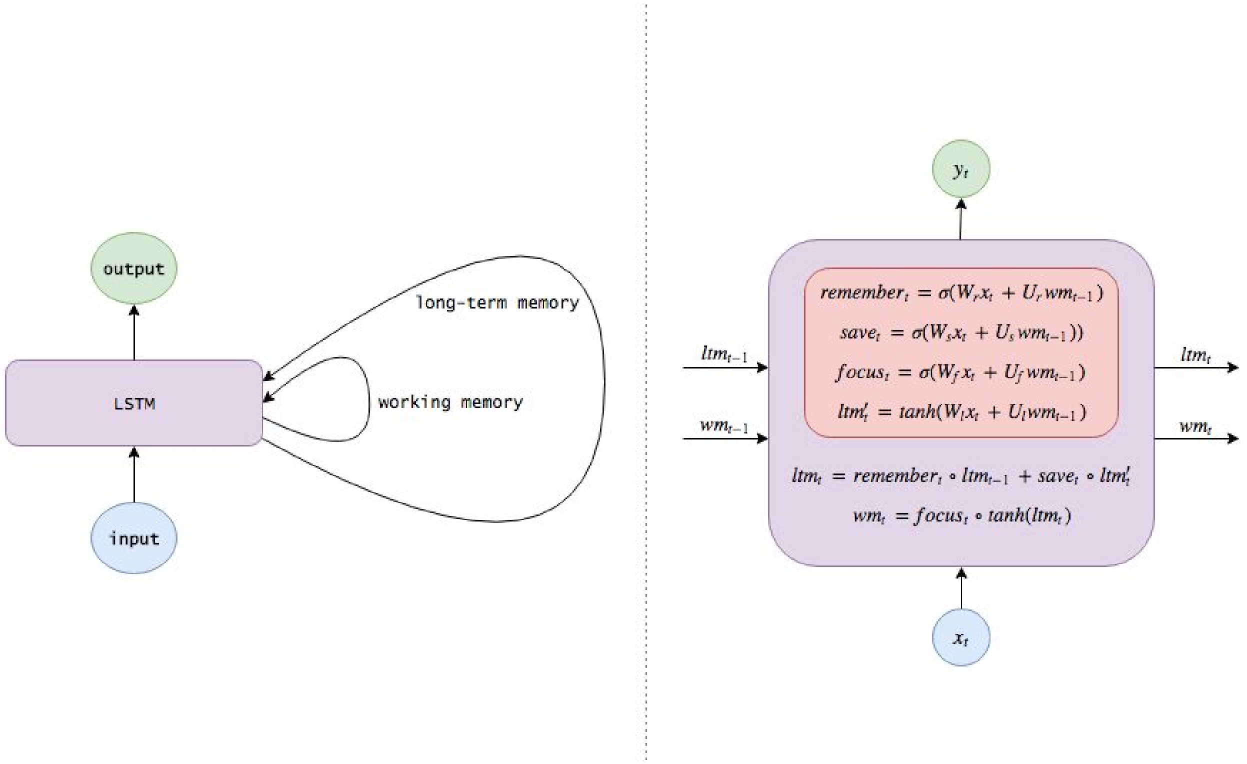
output
long-term memory
LSTM
working memory
input
y_t
ltm_{t-1}
wm_{t-1}
$remember_t = \sigma(W_r x_t + U_r wm_{t-1})$
$save_t = \sigma(W_s x_t + U_s wm_{t-1}))$
$focus_t = \sigma(W_f x_t + U_f wm_{t-1})$
$ltm'_t = tanh(W_l x_t + U_l wm_{t-1})$
ltm_t
wm_t
$ltm_t = remember_t \circ ltm_{t-1} + save_t \circ ltm'_t$
$wm_t = focus_t \circ tanh(ltm_t)$
x_t

Figure 2. An illustration of LSTMs on the left alongside an LSTM cell on the right featuring its mathematical representation, courtesy of Edwin Chen.

4 Investigation

Data collection

The dataset used in this paper comes from the publication "Automated Hate Speech Detection and the Problem of Offensive Language." It contains 24783 tweets, each labeled by a crowd of people as either hate speech, offensive language, or neither. The data was separated into training data (16,522 tweets) and validation data (8,261 tweets) with a 2:1 split. The training data is used to teach the model while the validation data is used to examine the accuracy of the model's predictions on data it has not seen.

Preprocessing

In the preprocessing step, unnecessary parts of the raw Tweet data are cleaned and removed. First, all links and username mentions (an @ followed by a Twitter handle) are deleted. Next, the tweet is split into an array where each word is a list element that is then converted into its respective token. Finally, using GloVe's pre-trained word embedding model, the texts are represented as vectors.

Model selection

The goal of the model is to learn the correlation between the Tweets and their classification (hate speech, offensive language, or neither) so that it can predict a classification for any given tweet. This task is considered a "supervised machine learning problem" as the model will train on a labeled data set. For machine learning applications that learn from textual data, advanced neural networks with an attention mechanism are commonly used. They are also ideal for this paper's application as they perform well on large data sets.

Model architecture

The model developed consists of a long short-term memory (LSTM) recurrent neural network (RNN). The input matrix fed into the model has a shape of 33 by 30 by 50 (batch size by the maximum number of words in a tweet by word vector dimension). The training data is passed into the model in batches with 33 data points in them, in order to train the model more efficiently. The output matrix has a shape of 33 by 3 (batch size by the number of classifications). The LSTM cell is made up of 64 units and is wrapped around a dropout wrapper with an output keep probability of 0.75. Rather than using gradient descent to learn the weights of the model, the Adam optimization algorithm is used.

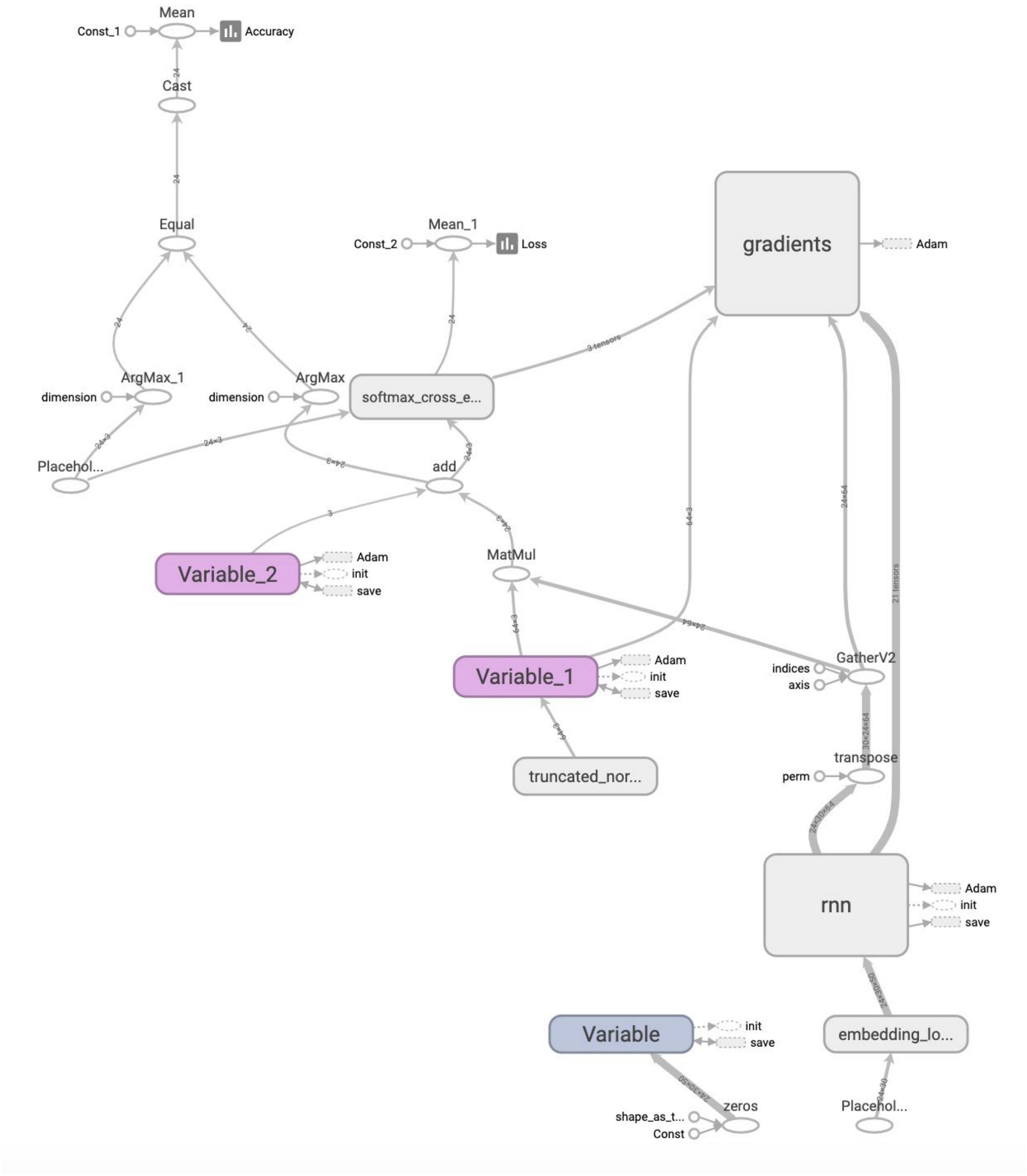

Figure 3. TensorBoard's visual representation of the model.

Model performance

The model was run for 100,000 iterations and resulted in high performance. This was repeated for multiple trials to ensure that the model was performing consistently. For every trial, the model was able to reach near 100% accuracy on the training data.

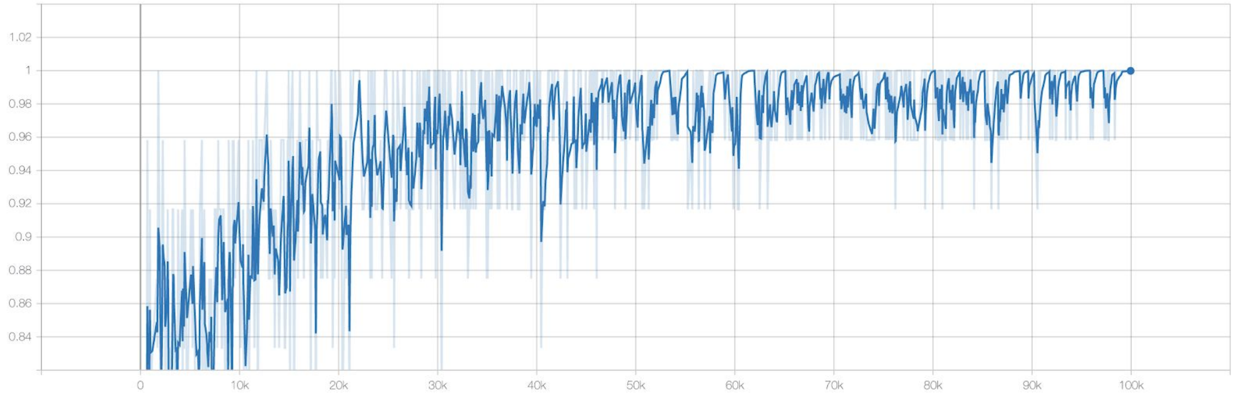

Figure 4. The graph of the model’s accuracy on the training data over time.

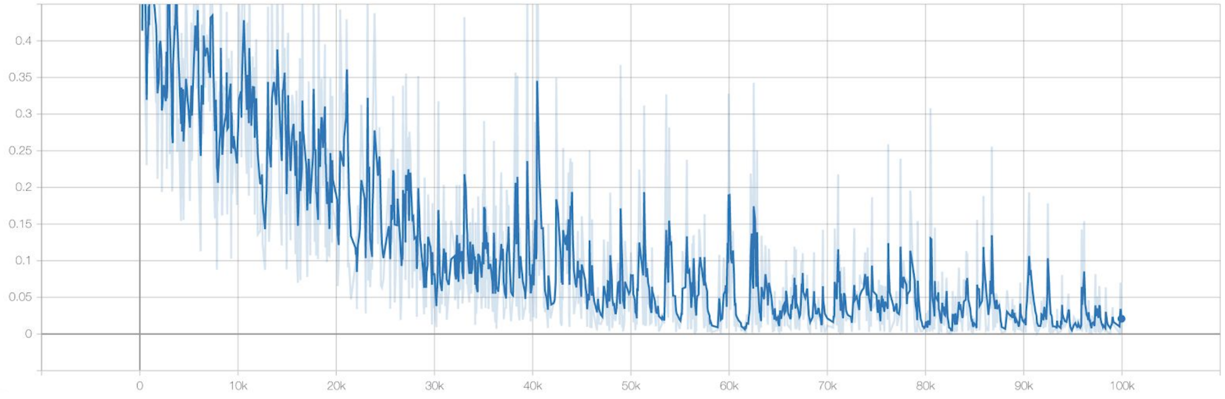

Figure 5. The graph of the model’s loss on the training data over time.

5 Conclusion

To classify a tweet as hateful, offensive, or neither, a model with a single LSTM layer was taught using a labeled dataset of tweets. After training the model on the training data for multiple trials, it was then tested against validation data to examine how well it performs on data it has not seen yet.

	Correctly Classified Hate Speech Tweets	Correctly Classified Offensive Tweets	Correctly Classified Neutral Tweets	Overall Accuracy on Validation Data
Trial 1	16.388888888%	90.838057300%	67.807720320%	83.7651331719%
Trial 2	18.611111111111%	90.592921709%	69.264384559%	83.9104116222%
Trial 3	17.50000000%	91.741994790%	67.370721048%	84.4552058111%
Trial 4	14.444444444%	90.516316837%	69.919883466%	83.777239709%
Trial 5	15.8333333333%	91.420254328%	69.1915513474%	84.430992736%
Average	16.55555556%	91.02190899%	68.71085215%	84.06779661%

Figure 6. Table of the model's accuracy on the validation data over multiple trials.

The average accuracy yielded on the validation data was ~84%. This means the model can fairly well classify tweets it has not seen yet as containing hate speech, offensive speech, or neither. The model was able to best identify offensive tweets, as it accurately predicted ~91% of offensive tweets in the validation set. The model also correctly predicted ~69% of tweets labeled as neither hateful or offensive in the validation set. Unfortunately, the model performed poorly when classifying hate speech and it only correctly predicted an average of ~17% of tweets labeled as hate speech in the validation data set, most likely as hate speech comes in various forms and the model could not easily determine patterns in tweets classified as hate speech. The accuracy could be improved by using a more appropriate pre-trained word-embedding model that includes the vocabulary found in tweets. Furthermore, other types of recurrent models could be examined to increase the model's accuracy.

6 Applications and Extensions

Applications

Predicting if a tweet is hateful, offensive, 0r neither can aid in alleviating the problem of hate speech on the Twitter social media platform. If the model predicts a composed tweet as hateful with high confidence, the tweet could be automatically removed from the platform. If the model is not as confident, the tweet could be flagged for a moderator to check and remove if deemed inappropriate. The model could also be applied to any other social media platform where users share textual content such as Facebook.

As well as filtering hateful content from social media platforms, the hate speech model could be implemented into child protected Internet browsers. A child protected Internet browser could run the model on a webpage and could restrict sites that contain hateful or offensive content.

Extensions

There are many other attentive recurrent models that could be trained to classify tweets as hateful, offensive, or neither. Rather than using a single LSTM layer as investigated in this paper, multiple LSTM layers could be stacked, possibly increasing performance. Bi-directional LSTMs could also be experimented with. Another type of recurrent long-term memory model is the gated recurrent units (GRU) model and could also be used to detect hate speech.

7 References

Atanasov, A. (2018). An Introduction to Recurrent Neural Networks [PowerPoint slides]. Retrieved from http://abatanasov.com/Files/Deep%20Learning%201.pdf.

Benioff, M. (2012, May 11). Welcome to the social media revolution. Retrieved from https://www.bbc.com/news/business-18013662.

Chen, E. (2017, May 30). Exploring LSTMs. Retrieved from http://blog.echen.me/2017/05/30/exploring-lstms/.

Colyer, A. (2016, April 21). The amazing power of word vectors. Retrieved from https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the 11th International AAAI Conference on Web and Social Media*, 512-515. Retrieved from https://arxiv.org/pdf/1703.04009.pdf.

Laub, Z. (2019, April 11). Hate Speech on Social Media: Global Comparisons. Retrieved from https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons.

Matsakis, L. (2018, September 26). To Break a Hate-Speech Algorithm, Try 'Love'. Retrieved from https://www.wired.com/story/break-hate-speech-algorithm-try-love/.

Nockleby, J. T. (2000). Hate Speech. In *Encyclopedia of the American Constitution* (pp. 1277-1278). New York, NY: Macmillan Reference USA.

Number of social media users worldwide 2010-2021. (n.d.). Retrieved from https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). doi:10.3115/v1/d14-1162

Terrell, K. (n.d.). The History of Social Media: Social Networking Evolution! Retrieved from https://historycooperative.org/the-history-of-social-media/.

Wang, C. (2018). Interpreting Neural Network Hate Speech Classifiers. *Proceedings of the Second Workshop on Abusive Language Online*, 86-92. Retrieved from https://www.aclweb.org/anthology/W18-5111.

What is Natural Language Processing? (n.d.). Retrieved from https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html.