FRI Natural language processing course 2021
Project report

University of Ljubljana
Faculty of Computer and
Information Science

Offensive Language Exploratory Analysis

Miha Petrišič, Veno Gaube, and Armin Komič

Introduction

Analyzing and monitoring hate speech has become of great importance for some social media platforms. For this task researchers have developed different techniques for identifying different types of hate speech. In this paper we will discuss some of the data and methods mostly used for analyzing and classifying different types of speech.

Our goal is first to analyze and visualize the data and then try to classify it accordingly to their class of hate speech only on the basis of the written text.

Methods

Long Short-Term Memory Neural Networks

Classificators based on neural networks have become more common when dealing with the classification of text. One of the more successful in the field is the Long short-term memory (LSTM) recurrent neural network (RNN). The networks are designed so they can effectively deal with the sequential data. That is why it has become popular with text based data.

Each neuron in a RNN takes it's output and returns it back as an input. By this, it adds additional information of the previous inputs of the network and it creates a kind of memory in the neuron. We can see how that is important in natural language processing since sequences of words are usually related. There, however, is a problem with remembering inputs that happened a while ago. In NLP we know that words can be far apart from each other, but still be correlated. That is where the LSTMs come in handy.

The LSTM solves the issue of long term memory with a cell state that helps to remember the values that happened in a certain time interval and uses three different gates that help to control the information that flows into cells.

Some researchers also used a modified version of the classic LSTM – Bidirectional LSTM, which helped produce even better results [1]. The idea behind this model is to create two LSTM one taking the input in the forward and one in the backward direction. That way we can preserve past as well as the future information. These sorts of models can usually understand the context of the text better than the classic LSTM.

Convolutional Neural Networks

Convolutional neural networks (CNN) are usually associated with computer vision. More recently however, they have been used to classify NLP problems as well. It is easy to understand how convolution works on images, but applying convolution to text might at first seem a bit odd, but with proper preprocessing of the data we can represent our documents or sentences as a matrix. Typically we use word embeddings to represent words as vectors. Each of the vectors represents a row in a matrix. We then use filters with a width of a word vector and a height of usually about 2-5 words to convolve the matrices.

However CNNs have problems with making connections with words that are far from each other. In images the pixels that are close to each other are usually correlated, in a text that is not always the case. That's why RNNs usually make more sense for NLP tasks. That being said, the CNNs are very fast and have had usage in different tasks regarding hate speech detection [1, 2].

Support Vector Machines (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm capable of performing classification, regression, and even outlier detection. The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space(N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, many possible hyperplanes could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

In the paper about hate speech dataset from a supremacy forum [3], SVN algorithm was used over Bag-of-Words vectors for developing a custom hate speech annotation tool, and results of 74% accuracy were achieved.

Word2vec

Word2vec [4] is a technique for natural language processing. The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. It represents each distinct word with a particular vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.

Word2vec is a group of related models that are used to produce word embeddings.These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec [4] takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.

Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TFIDF) [5] is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Term frequency

The number of times a term occurs in a document is called its term frequency [5]. Suppose we have a set of English text documents and wish to rank them by which document is more relevant to the query, ”the brown cow”. A simple way to start out is by eliminating documents that do not contain all three words ”the”, ”brown”, and ”cow”, but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document.

Inverse document frequency

Because the term ”the” is so common, term frequency will tend to incorrectly emphasize documents which happen to use the word ”the” more frequently, without giving enough weight to the more meaningful terms ”brown” and ”cow”. The term ”the” is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less-common words ”brown” and ”cow”. Hence, an inverse document frequency [5] factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Features

LIWC Feature (Linguistic Inquiry and Word Count)

The way that the Linguistic Inquiry and Word Count (LIWC) program works is fairly simple. It reads a given text and counts the percentage of words that reflect different emotions, thinking styles, social concerns, and even parts of speech. The LIWC program includes the main text analysis module along with a group of built-in dictionaries. Each word is converted into a 125 dimension LIWC vector, one dimension per semantic category.

In the paper describing hate speech detection using context-aware models [6], the performance of logistic regression models was improved when LIWC Features were added to the feature set.

NRC Emotion Lexicon Feature

The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing. The NRC Lexicon has affect annotations for English words. Despite some cultural differences, it has been shown that a majority of affective norms are stable across languages.

In the paper describing hate speech detection using context-aware models [6], NRC Emotion Lexicon was used to capture emotion clues in the text. Each word was converted into a 10 dimension emotion vector, corresponding to eight emotion types and two polarity labels. The emotion vector for a comment or a sentence is a 10 dimension vector as well, which is the sum of all its words’ emotion vectors. In combination with LIWC and other features, NRC features also added slightly improved final results.

Datasets

List of usable offensive language datasets that are feasible to retrieve and their associated papers:

- **Automated Hate Speech Detection and the Problem of Offensive Language** [7] - data divided into Hate, Offensive and Neither classes.

- **Hate Speech Dataset from a White Supremacy Forum** [3] - data divided into Hate, Relation and Not classes.

- **Detecting Online Hate Speech Using Context Aware Models** [6] - data annotated with Hate and NotHate classes.

- **CONAN - COunter NArratives through Nichesourcing: a Multilingual Dataset of Responses to Fight Online Hate Speech** [8] - data annotated with Islamophobic and NotIslamophobic classes and Culture, Economics, Crimes, Rapism, Terrorism, Women Oppression, History, Other subclasses.

- **A Benchmark Dataset for Learning to Intervene in Online Hate Speech** [9] - data annotated with Hate and NotHate classes.

- **Exploring Hate Speech Detection in Multimodal Publications** - [10] every text annotated by three different annotators. Classified in 6 different categories: no attacks to any community, racist, sexist, homophobic, religion based attacks and attacks to other communities.

- **Peer to Peer Hate: Hate Speech Instigators and Their Targets** [11] - data divided by classes ”hate” and ”not”, only for tweets which have both a hate instigator and hate target.

- **Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages** [12] - data divided into three branches of classes: A: hate / offensive or neither, B: hatespeech, offensive, or profane, C: targeted or untargeted.

References

[1] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[3] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*, 2018.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. 3(1), 2013.

[5] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal*, 1957.

[6] Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*, 2017.

[7] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

[8] Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. Conan–counter narratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. *arXiv preprint arXiv:1910.03270*, 2019.

[9] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. A benchmark dataset for learning to intervene in online hate speech. *arXiv preprint arXiv:1909.04251*, 2019.

[10] R. Gomez, J. Gibert, L. Gomez, and D. Karatzas. Exploring hate speech detection in multimodal publications. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1459–1467, 2020.

[11] Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. Peer to peer hate: Hate speech instigators and their targets, 2018.

[12] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. New York, NY, USA, 2019. Association for Computing Machinery.