Sentiment Classification on Reddit comments on Climate Change

Jaskirat Marar
732A81 - Text Mining
jasma356@student.liu.se

Abstract

This project takes the form of an investigative study in analyzing user comments posted on Reddit to determine the user's sentiment on Climate Change. The study proposed to compare the classification capabilities of simple RNNs to learn semantics & syntactic form of the comments versus a LSTM that captures the long-term dependencies using word embeddings from pre-trained GloVe vectors. The importance of context becomes apparent immediately when we combine word embeddings from GloVe with a LSTM which results in a huge boost with an AUC of 80% vs just a 57% AUC score for a simple RNN. We attempt to improve upon the performance of the simple RNN by replacing it with a GRU and use the word embeddings from GloVe to receive an AUC score of 84%.

1 Introduction

Text classification is one of the earliest use cases of NLP and there are numerous supervised and unsupervised ML algorithms like Logistic Regression, Support Vector Machines, Decision tree methods, Ensemble methods, clustering etc that are employed widely for classification tasks. Where traditional classifiers have always lacked is in understanding context. That is where NNs, word embeddings and transformers have come into play in the evolution of NLP as a research area. The primary reason for the limitations of traditional ML classifying methods lies in fact that they are built upon word representations and generalize on training data made available in the form word vectors. Hence, the next generation of modelling methods fixes this by utilizing similarities between words as a distance or angle to capture deeper relationships between words and capture contextual relevance Maas et al. (2011). In this project I have established the limitations of traditional ML algorithms to plateau in their ability to generalize and classify a sentiment label based on word representations. After I establish this, I introduce the capture of contextual relationship to improve the performance of the classification task by making use of word embeddings and deep learning methods to learn temporal dependencies in user posts. One of the drawbacks of working with social media user posts is that the posts are small in length and often written as short form opinions of users without proper motivations and explanation Wang et al. (2018). This creates a challenge in limiting the capability to learn contextual associations. To address this, I make use of pre-trained word embeddings to make up for the limitations of learning from the given dataset.

2 Related Literature

2.1 Word Embeddings

To successfully accomplish a sentiment classification task, we need to understand and capture the semantic compositionality of the data Sag et al. (2002). The most popular advances in this field of NLP have been made using word embeddings. Word embeddings derive the relationship between words by using the cosine distance between their vector representations. Similar words have their vector representations also closer to each other in terms of cosine distance.

Therefore embeddings for phrases can also be built by adding the individual vectors of the words in the phrase. Two of the most popular algorithms for obtaining word embeddings are word2vec and GloVe. Both the models are different in how they are trained. Word2vec is FFN which is either modelled on the skip-gram model or the continuous bag-of-words or CBOW model. GloVe is matrix factorization technique which constructs large matrices od co-occurrence (words x context) which is then factorized to a lower dimension matrix (words x features) in which each row in the matrix gives the

vector representation of the corresponding word. Pennington et al. (2014) have established GloVe to be the superior model for the unsupervised learning of word representations which outperforms other models like word2vec in word analogy, similarity and NER tasks.

2.2 Neural Networks Architecture

Recurrent NNs came into the field of NLP to overcome the barrier of learning temporal dependencies between words that give contextual meaning to the sentence. This was never possible by using the traditional ML classifying techniques which rely on generalizing through training on word vectors. Contextual word representations are derived from Bi-directional Language models (Bi-LMs) which combine a forward and backward language model to jointly maximize the likelihood in both directions Peters et al. (2018):

$$\sum_{k=1}^{n}(logP(t_k|t_1,...,t_{k-1};\overrightarrow{\Theta}) + logP(t_k|t_1,...,t_{k-1};\overleftarrow{\Theta}))$$

One of the most popular RNN variant is the LSTM (long-short term memory). They have been proven to provide state of the art performance for several language modelling tasks and they work by allowing for connections between units which unlike FFNs allows them to use their interna states to process sequences of inputs and capture temporal dependencies. Another popular variant of RNNs is GRU or Gated Recurrent Unit which is similar to a LSTM but without the output gate Chowdhury and Zamparelli (2018). GRU was introduced by Cho et al. (2014) and it aims to solve the vanishing gradient problem that occurs with standard RNNs. Because of the lacking output gate, the memory cell contents are copied into the network at each time step. GRUs are supposed to be simpler and faster than LSTM and generally produce par results if not better than LSTMs.

3 Data

I experimented on a reddit comments dataset made publicly available on Kaggle. The data set contains the posts and comments on Reddit which mention the terms "climate" and "change". The data is quite an extensive corpus of comments with comments scraped till 2022-09-01 and contains nearly 40M comments. The dataset has been limited to include data about subreddit, NSFW(Not Safe For Work) tag, comment date, comment text, link to the comment, the score of the comment (upvotes/downvotes) and an analyzed sentiment for the comment. The dataset was chosen because it had a sentiment score which could be converted into classes. This helped me in the evaluation of the models. It was not clarified by the author as to what methodology had been used in calculating the sentiment scores. But a sample check was sufficient to infer that the sentiment score reflected the true sentiment label of the comments. In the interest of computational resources available to me, I have chosen to use only 1% of the dataset which still amounts to a corpus of more than 40000 comments.

4 Method

4.1 ML Algorithms

Ignoring any underlying contextual relationships within sentences and phrases, any sentiment analysis is at its core, simply a text classification problem. In our case, because the data is labelled, this can be modelled as a supervised text classification problem with the intention to showcase the limits of using traditional classification algorithms. For this effect we setup and train the following ML & ensemble classifiers to identify a baseline as well as to estimate what is the best classifier we can train to give us the best prediction accuracy.

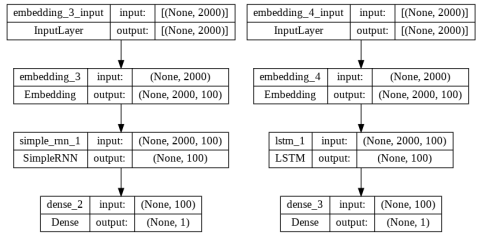
Classifier	Params
Naive Bayes	default
Logistic Regression	max_iter=1000
SGD	tol = 10^-3
Random Forest	100 trees
AdaBoost	100 estimators, lr = 0.9
XGBoost	100 estimators, lr = 0.1, depth = 10

The above classifiers help in establishing a baseline of nearly 70% with the Naïve bayes. But inspecting the other classifiers we realise that the we hit a ceiling of prediction accuracy at about 73-75% for ensemble classifiers and we can reach about 80% accuracy with the gradient based classifiers.

4.2 RNNs

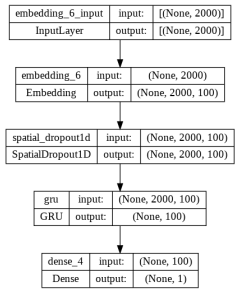
After establishing this baseline and plateau performance, we then move to the main idea of this project i.e. employing word embeddings for semantic relations and finding a model that best classifies the sentiment of the user opinions. To this effect we will use the following 3 RNN variants:

RNN Variant	Pre-trained Embeddings
Simple RNN	None
LSTM	Glove-twitter-27B.100d
GRU	Glove-twitter-27B.100d



(a) Simple RNN without pre-trained embeddings

(b) LSTM with GloVe embeddings



(c) GRU with GloVe embeddings

Figure 1: RNN Architecture for chosen Models

The architecture of each of these variants is given in the figures. We first use a simple RNN without any pre-trained embeddings and one dense layer. We follow this up by constructing a LSTM model and another model with GRU. In both these models we use pre-trained embeddings from GloVe vectors. For this project, I tried with 2 different embeddings, first the 6Bn tokens trained on Wikipedia2014 + Gigaword5 and secondly, the vectors from 27Bn token trained on 2 Bn tweets. I found the results to be slightly more favorable with the Twitter based embeddings which makes sense since my dataset is also from social media. This pretrained embeddings were used to construct an embedding matrix for the vocabulary. I use this matrix as weights for the embedding layer in the LSTM and the GRU models.

5 Results

5.1 Evaluation Criteria

Since the dataset is balanced in terms of sentiment counts with an equal representation of positive and negative sentiment, we compare the models using AUC scores. The label classification we will get from our trained NNs will give us probabilities to work with hence, the choice for using AUC to evaluate & compare model performance.

Classification Results	AUC Score	f1 Score
Naive Bayes	0.69	0.69
Logistic Regression	0.80	0.81
SGD	0.79	0.80
Random Forest	0.73	0.74
AdaBoost	0.73	0.73
XGBoost	0.74	0.74

We can infer from the results of the ML classification algorithms that the best performance can be obtained from training a logistic regression classifier which yields an AUC score of 80%. Ensemble methods don't perform as well as the gradient descent methods. We move now to the contextual language modelling based RNNs.

RNN Variant	AUC Score	Accuracy
Simple RNN	0.57	0.55
LSTM	0.80	0.73
GRU	0.84	0.76

We see the impact of using pre-trained embeddings with both LSTM and GRU NNs giving an AUC score of 80% or more. The best performance was obtained with the GRU with an overall AUC score of 84%. By increasing the number of epochs we are able to improve the AUC score to 87%.

6 Discussion

In this project I have been able to show that to tackle a sentiment classification problem we cannot rely alone on word vector based ML classifiers. It will result in a plateaued performance and we will not be able to improve the performance of our classifier beyond a certain point. The importance of contextual language modelling is established by the use of RNNs and word embeddings. We try to learn word embeddings without using any pretrained set which results in a very low scoring model. This was expected because learning word embeddings would require a huge training corpus and very steep computational resources. That is why we make use of pretrained GloVe vectors which have been trained on twitter tweets. We immediately see a boost in AUC scores.

Our best performing model was the Gated Recurrent Units model. The benefits of GRUs are manifold, the biggest being the speed of training

due to its ability to deal with the vanishing gradient problem. Due to slow training times Icould only test the LSTM with upto 3 epochs. Because the GRU was so faster to train I could train with upto 15 epochs and get the best AUC score among all the models. Any attempts to train on more training data were also unsuccessful.

7 References

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Conference on Intelligent Text Processing and Computational Linguistics*.

Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, and Long Wang. 2018. An LSTM approach to short text sentiment classification with word embeddings. In *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018)*, pages 214–223, Hsinchu, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).