Stance classification in tweets using Recurrent Neural Network

Ayaz Aziz Mujawar (s3751555), Pranamya P Korde (s3779009)

Abstract

Stance classification is a subcategory of opinion mining where the task is to automatically determine whether the author of a piece of text is in favor or against a given target. It can be formulated in different ways. In our context, we define stance detection to mean automatically determining from the text whether the author is in favor or against of the given target, or whether neither inference is likely (Saif M Mohammad, 2016).To successfully detect stance, automatic systems often have to identify relevant bits of information that may not be present in the focus text (Saif M Mohammad, 2016). Automatically detecting stance has widespread applications in information retrieval, text summarization, and textual entailment.

This is a project where we have implemented a Deep Learning technique to automatically detect the stance of a particular tweet, when the tweet statement and target of that tweet is provided for the data taken from SemEval – 2016 Task 6 competition. The source of a data is "Semeval-2016 Task 6: Detecting Stance in Tweets. Saif M. Mohammad et. al. In Proceedings of the International Workshop on Semantic Evaluation (SemEval-16). June 2016". The main aim of our project is to build a deep learning model to classify the type of stance from tweets that are associated with one of five politically-charged targets: "Atheism", "the Feminist Movement", "Climate Change is a Real Concern", "Legalization of Abortion" and "Hillary Clinton".

Introduction

We can often detect from a person's utterances whether he/she is in favor of or against a given target entity— their stance towards the target. However, a person may express the same stance towards a target by using negative or positive language. So, the given dataset consists of tweet–target pairs annotated for both stance and sentiment. The targets may or may not be referred to in the tweets, and they may or may not be the target of opinion in the tweets. Partitions of this dataset were used as training and test sets in a SemEval-2016 shared task competition (Saif M. Mohammad, 2016). The data we have is separated into two datasets for training and testing purpose according to the timestamp of the tweet (Saif M Mohammad, 2016).The stance annotations are expressed using 3 classes namely 'FAVOR', 'AGAINST' and 'NEITHER'. In this experiment, we used two different approaches to build a solution and evaluate them to check if we can automatically detect and predict the Stance class if the tweet statement and its target is provided to the automated deep learning model. This report consists of detailed explanation of design, analysis and evaluation of our methodology, investigation, and results of this experiment.

Literature Review

Many researchers in the field of Natural Language processing has directed broad research for detecting the sentiments and stances from the texts. (Gayathri Rajendran, 2018) demonstrated the system that detects the stance of the user from the texts. The dataset collected for this paper consists of news channel articles. The dataset consists of "news headlines", and its "bodies" associated with each headline. Each headline is associated with multiple bodies and correct relationship between "headline" and "bodies" are paired which is considered as a target feature ("agree", "disagree", "discuss", and "unrelated"). Initially the dataset is divided into 70% of training set and 30% of validation set. Data is pre-processed using techniques like tokenization and stop words removal. Word Embedding are learned and applied on headline and body text, and also those texts are then vectorized. These vectors of dimensions 200 is given as an input to different deep learning models. The evaluation metric selected for this case is accuracy and out of all, Bi-directional LSTM outperforms all other models with the accuracy of 83.5%.

(Elena Kochkina, 2017)determines the rumor veracity and support for the rumours. The dataset collected in this paper consists of twitter conversation thread connected with rumours, and associated target labels are divided into 4 sections (Support, Query, Deny, and Comment). The tweets are pre-processed by removing non-alphabetic characters, converting words to lower case and by tokenizing the texts. The features are extracted from the text using word2vec and tweet lexicons, and then they are converted into vectors which are then fed as an input to the branch LSTM network. The dataset is divided into training, development, and test set and hyperparameter tuning is done experimentally to improve the performance of the model. Along with Accuracy, macro-average F score is also considered as an evaluation metric for this experiment. There is not much difference between development and test set for accuracy metric but there is a significant difference between the macro-averaged F-score due to class imbalance. So, authors decided to go with the accuracy metrics as they achieved 78.40% on the test set.

(Guido Zarrella, 2016)worked on the same dataset that provided to us i.e. SemEval-2016 Task 6 consists of 2814 tweets which belongs to 5 independent topics. The input tweets first converted to tokens and then this token is one hot encoded and pretrained using weights of 256 dimensions which were initialized using word2vec skip gram algorithm. Then these sequences of inputs are fed to the Recurrent Neural Network with 128 LSTM units. The output of this model is densely connected with the RELU along with 90% dropout and this layer input is fed to the SoftMax layer which helps to represent output in multiclass format. The model is tested on 1249 holdout tweets labels. F1-score was considered as an evaluation metric for this task and they achieved overall F1-score of 67.8%.

(Jiachen Du, 2017) developed a solution in which they proposed a neural attention model which was used to extract the text as well as target related information for detecting the stance. Authors of this paper designed RNN based model along with Bi-directional LSTM which can concentrate on important parts of the text based on the target. The datasets for the experiments are taken from semeval-2016 competition. Target augmented embedding was used to combine the text tweets with the target. Micro-average F1-score is considered as an evaluation metric for this experiment. Ad-hoc strategy was used in which one model for each target is trained and final results are obtained by concatenating all the predictions. Hyperparameter tuning along with 5-fold cross validations are applied on the training set. While performing the experiments, word2vec was used to convert text into vectors and pretrained word embedding vector are crawled from twitter and sina microblogging. The author concludes that target specific attention models outperforms all other baseline models with the F1-score of 68.79%.

METHODOLOGY

Data Exploration and Preprocessing .

SemEval 2016 dataset for Task 6 is divided into 2 files for training and testing purpose, respectively. Training dataset consists of 2914 tweets along with their targets and annotations like ‘Stance’, ‘Opinion towards’ and ‘Sentiment’ while test data is comparatively small with 1955 tweets. Tweets are related to only five target subjects: ‘Hillary Clinton’, ‘Feminist Movement’, ‘Legalization of abortion’, ‘Atheism’, ‘Climate Change is Real Concern’. And their Stance values are divided into 3 classes ‘FAVOR’, ‘AGAINST’ and ‘NONE’. After exploring the training dataset, we came to know that there is a tweet imbalance between these 5 target classes. That means, there are 689 tweets present for target ‘Hillary Clinton’ while there are only 395 tweets available in the data for target ‘Climate Change is a Real Concern’. We also observed the problem of class imbalance across the training dataset. There are 1395 tweets which are in AGAINST of their respective targets while only 753 tweets are in FAVOR of the target. There are 766 tweets which are neither in FAVOR nor AGAINST the target subject. Moreover, we observed that there is a great imbalance in Stance opinions between Target values. For example, there are 393 tweets present in the training data which are against target ‘Hillary Clinton’ but there are only ’15’ tweets present which are against ‘Climate Change is a Real Concern’. In the below graph (Fig: 1.1), we can see the class imbalance problem present in the data.

As we can see the varied class imbalance across each topic, some topics (e.g. ‘Climate Change is a Real Concern’) show significant skew while other are comparatively more balanced. Because of this imbalance performance metric like Accuracy will not yield a true result while developing a model. Therefore, we have used F1-score with weighted average as a performance measure for evaluating our solution. The weighted average F1-Score takes class imbalance into consideration. As we are considering all three classes into consideration while evaluating the model we have used ‘Categorical Crossentropy’ as our error metric and weighted F1-score as our performance measure.

We have to do some more exploration to check out some more facts about the data we have. This include, exploring the word count, average character count, stop word count, number of hashtags (#) and number of mentions (@), number of email IDs and number of digits present in the data. As a preprocessing part, we perform multiple actions to make tweet statements better for analysis and modelling. This preprocessing includes, converting data to lower case, removing the contractions in the words, removing email ids and other extra characters except (# and @), removing accented characters and stop words from the tweets and removing extra white spaces from the tweets. We concatenate the tweet statement with their associated topic target value to make a single string for each tweet which contain topic value as well. We tokenize and pad the tweet data with maximum length of 50. To make it a multiclass classification problem, we encode the Stance value with simple one hot encoding.

MODELLING

Methodology and System Overview .

We implemented 2 different approaches to stance detection which employs recurrent neural network organized into 4 layers of weights as mentioned in the system architecture figure. Both approaches use similar system architecture and they do not incorporate any manually engineered task-specific features or inputs relevant to the surface structure of the text. The only inputs to the network were the sequence of indices representing the identity tokens (words or phrases) in the text. In the first approach we use all the input tweets to feed to the single system and evaluating the results and in second approach we train 5 different models each for one topic and averaging their performance will give us overall result of the experiment. The detailed explanation network architecture is given below.

System Architecture .

The system designed consists of 4 weight layers. Input tokens are encoded such that each token is sparse vector. Sequence input of these vectors are projected through 100 dimensional pretrained embedding layer which feeds into a recurrent layer containing of 32 Long Short-Term Memory (LSTM) units. This LSTM units have dropout rate of 0.5 and recurrent dropout rate of 0.25. The terminal output of recurrent layer is densely connected to the 64-dimensional Rectified Linear Unit (ReLU) layer with dropout of value 0.6. Finally, the output of this layer is fully connected to the 3-dimensional softmax layer in which each unit is represents one of the output classes: FAVOR, AGAINST, or NONE.

Transfer Learning with pretrained vector:

Transfer Learning is an approach where knowledge learned from a particular network is used for training our network by using the weights learned previously. This will improve our vocabulary representation by mapping all the words from our vocabulary to the words represented in the pretrained network. There are two main libraries used for this purpose GloVe and word2vec.

The preprocessed and encoded data passed to the pretrained embedding layer. This embedding layer uses weights of the pretrained vector downloaded externally from https://nlp.stanford.edu/projects/glove/. We have used 100-dimensional twitter pretrained vector which contains 27B token of almost 1.2M vocabulary. We used these weights in the embedding layer to perform transfer Learning. Now to avoid this embedding layer again from training, we put trainable parameter as False in the embedding layer.

(a.) Approach 1

In this approach we have used all the training data points to train and evaluate a single deep network. We used the similar network architecture as above.

(b.) Approach 2

In the second approach, we consider the imbalance of data between different topics and hence we divide the whole dataset into 5 subsets each for 1 topic. We developed a similar network as above for each of these topics and try to evaluate the performance of the model for each topic. Averaging the performance of each of the 5 models will give us the overall result of the experiment. While testing the performance of this approach on the test data, we have to divide the test data according to the topics and then feed it to the respective model to check the performance of the model on the unseen data. We checked the approach 2 performance with network with and without transfer learning applied.

EVALUATION

Experiment and Performance Tuning .

As mentioned above, we have implemented 2 different approaches to classify the stance class using the deep network architecture mentioned above. In the first approach, first we initialize the embedding layer with the weights obtained by the pretrained vector mapping (transfer learning). The network is trained with 'adam' optimizer with default learning and decay rate. The network is compiled using Categorical cross-entropy loss function. The recurrent networks are implemented using Keras framework. The training data was shuffled and split into training and validation sets using simple 'train_test_split ()' method from sklearn library. We use 80% of the total data for training and remaining 20% data for validating the model and to check how well the model is generalizing. The network is trained for 50 epochs and the batch size is set to 64. In the second approach we implemented 5 different classifiers each for one topic and the training data is split into 5 subsets. For each subset, we use 80% of the subset data to train the corresponding network. Each network is compiled with same 'adam' optimizer and Categorical cross-entropy loss function. Each network is trained for 50 epochs with batch size of 32.

We have checked the other variants of these approaches. In the approach 2 we also omitted the pre-trained embedding layer and try using learn from scratch approach. These variants were not found to improve the performance. The next thing we checked is using the 3-fold cross validation for training. But it is resulting in high training performance but low test-data performance.

At the start of performance tuning, we referred the network created by MITRE's submission for SemEval 2016 competition as they have used similar RNN structure with transfer learning and their network won highest average performance score on the same dataset. We used the same parameter values but as we are using GloVe pretrained vector for initializing the pre-training layer weights and we don't have enough data to manage such a large pretraining layer, the performance was quite low. So, we tried combinations of different values for LSTM memory units, Dropout rates, Dense layer dimensions and regularizer values. We also tried using bidirectional LSTM network.

We found out the maximum score is achieved using below parameter values.

LSTM memory units = 32

ReLU units = 64

Dropout rate = (0.5, 0.6, 0.4)
Recurrent dropout of = 0.25
We even try training the network using different optimizers like Stochastic Gradient Descent (SGD with learning rate of 0.015 and momentum of 0.9) which gives us similar results as that with using 'adam' optimizer.

Results .

Our approach one achieved weighted average F1 score of 49.29 on the preprocessed and tokenized test data. From Fig:2 we can observe decline in overfitting due to decrease in loss. While average F1 score achieved by approach 2 is different for each of the 5 models which is ranging from 43.73 to 71.57. There are variations between the performances of each model and some models are overfitting by slight margin.

Independent Evaluation .

Independent Evaluation on the dataset is performed using approach 1. This dataset consists of 11 tweets along with their target and stance. We evaluated our model on this dataset and try to find out if our model is performing as expected n the unseen data further. Our model performed well on this dataset achieving F1 score of around 57.14.

Ultimate Judgement and Limitation .

We described 2 different approaches to automatically determining the stance of an author based on the content of a single tweet. These approaches make use of Transfer Learning and weights learned by the pre-trained vectors to maximize the performance of the model in some cases. We faced some limitations like imbalance in classes across each target topic and limited amount of data for training. But according to the (Saif M. Mohammad, 2016) our score lies within performance benchmark set by most of the classifiers. Transfer learning does not completely eliminate the need of labeled in-domain training. Future experiments could investigate other methods and approaches to solve this classification problem having above limitations.

Appendices

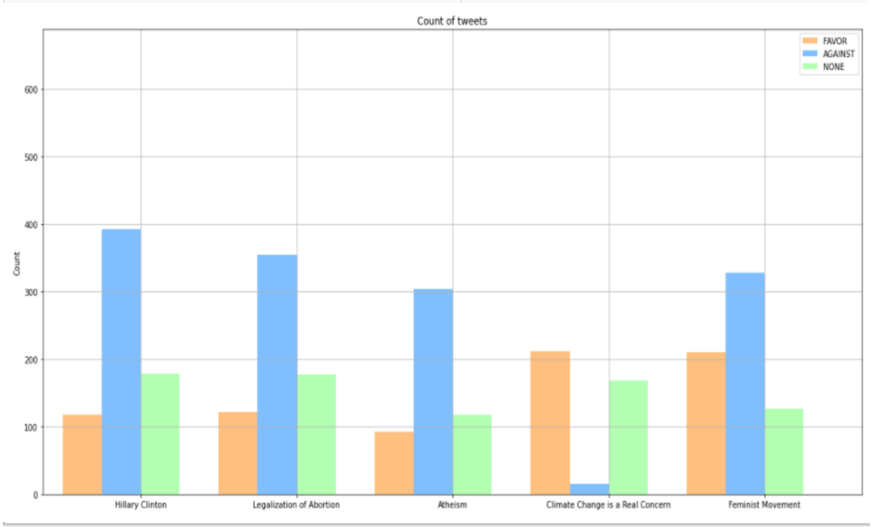

Fig: 1 Data exploration of class distribution for each target topic

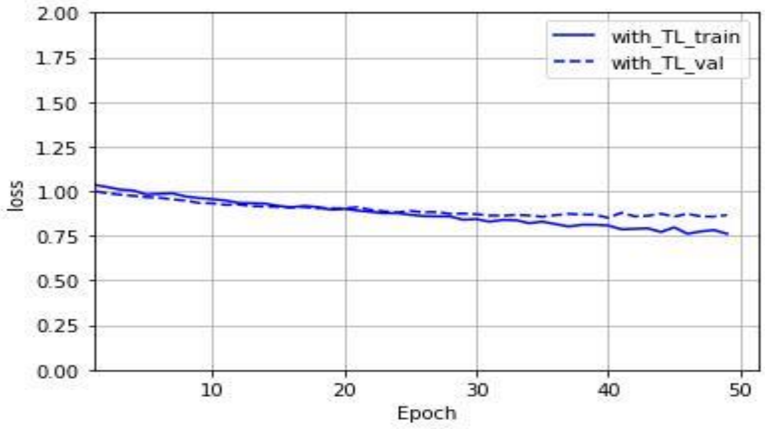

Fig 2: Learning curve of approach 1

Model Name	F1-Score
Approach1_All_data_model	0.492988
Approach2_HC_WTF	0.531707
Approach2_HC_TF	0.624506
Approach2_AB_WTF	0.544041
Approach2_AB_TF	0.554545
Approach2_AT_WTF	0.600639
Approach2_AT_TF	0.715789
Approach2_CC_WTF	0.616740
Approach2_CC_TF	0.637555
Approach2_FM_WTF	0.437376
Approach2_FM_T	0.661123

Table 1.1: F1-score of all the models

REFERENCES

[1]Elena Kochkina, M. L. (2017). Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), 475-480.

[2] Gayathri Rajendran, B. C. (2018). Stance-In-Depth Deep Neural Approach to Stance Classification. International Conference on Computational Intelligence and Data Science, 1646-1653.

[3] Guido Zarrella, A. M. (2016). MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. arXiv:1606.03784v1 [cs.AI].

[4]Jiachen Du, R. X. (2017). Stance Classification with Target-Specific Neural Attention Networks. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).

[5]Parinaz Sobhani, S. M. (2016). Detecting Stance in Tweets And Analyzing its Interaction with Sentiment. Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, 159–169.

[6] Saif M Mohammad, S. K. (2016, 01 11). http://alt.qcri.org/semeval2016/task6/. Retrieved from http://alt.qcri.org/semeval2016/task6/: http://alt.qcri.org/semeval2016/task6/

[7] Saif M. Mohammad, P. S. (2016). Stance and Sentiment in Tweets. ACM Transactions on Embedded Computing Systems, Vol. 0, No. 0, Article 0, Publication date: 2016.