STANCE DETECTION WITH BIDIRECTIONAL CONDITIONAL ENCODING

PROJECT REPORT – CS529

Submitted to

Dr. Sanasam Ranbir Singh

Submitted by

Shubham Sachan-214161010

Nivedita Gaikwad-214161015

Ankit Raj-214161018

Ashish Dev Bharadwaj-214161023

MTech Data Science (2021-2023)

Indian Institute of Technology, Guwahati

PHASE – I:

Introduction

The goal of stance detection is to classify the attitude expressed in a text towards a given target, as “positive”, “negative”, or “neutral”. Such information can be useful for a variety of tasks. In this report we focus on a novel stance detection task, namely tweet stance detection towards previously unseen targets (mostly entities such as politicians or issues of public interest), as defined in the SemEval Stance Detection for Twitter task.

Background of Research Problem

Stance detection is the task of classifying the attitude. Previous work has assumed that either the target is mentioned in the text or that training data for every target is given.

This paper considers the more challenging version of this task, where targets are not always mentioned and no training data is available for the test targets. We experiment with conditional LSTM encoding, which builds a representation of the tweet that is dependent on the target, and demonstrate that it outperforms encoding the tweet and the target independently.

Challenges in Research Problem:

Unseen target stance detection :-

This task is rather difficult, firstly due to not having training data for the targets in the test set, and secondly, due to the targets not always being mentioned in the tweet.

For example, the tweet “@realDonaldTrumpis the only honest voice of the @GOP” expresses a positive stance towards the target Donald Trump. However, when stance is annotated with respect to Hillary Clinton as the implicit target, this tweet expresses a negative stance, since supporting candidates from one party implies negative stance towards candidates from other parties. Thus the challenge is twofold. First, we need to learn a model that interprets the tweet stance towards a target that might not be mentioned in the tweet itself.

Second, we need to learn such a model without labelled training data for the target with respect to which we are predicting the stance. In the example above, we need to learn a model for Hillary Clinton by only using training data for other targets.

While this renders the task more challenging, it is a more realistic scenario, as it is unlikely that labelled Training data for each target of interest will be available.

Paper proposed model

In the model proposed by paper three approaches were used

1. Independent Encoding
2. Conditional Encoding
3. Bi-directional Conditional Encoding

1. Independent Encoding

Our initial attempt to learn distributed representations for the tweets and the targets is to encode the target and tweet independently as k-dimensional dense vectors using two LSTMs.

$$
\begin{aligned}
\mathbf{H} &= \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \\
\mathbf{i}_t &= \sigma(\mathbf{W}^i\mathbf{H} + \mathbf{b}^i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^f\mathbf{H} + \mathbf{b}^f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o\mathbf{H} + \mathbf{b}^o) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}^c\mathbf{H} + \mathbf{b}^c) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}
$$

Here, x_t is an input vector at time step t, c_t denotes the LSTM memory, h_t € R^k is an output vector and the remaining weight matrices and biases are trainable parameters. We concatenate the two output vector representations and classify the stance using the softmax over a non-linear projection

$$\text{softmax}(\tanh(W^{ta}\, h_{target} + W^{tw}\, h_{tweet} + b))$$

into the space of the three classes for stance detection where W^{ta} , W^{tw} € $R^{3\times k}$ are trainable weight matrices and b € R^3 is a trainable class bias. This model learns target-independent distributed representations for the tweets and relies on the nonlinear projection layer to incorporate the target in the stance prediction.

2. Conditional Encoding

In order to learn target-dependent tweet representations, we use conditional encoding. We use one LSTM to encode the target as a fixed-length vector. Then, we encode the tweet with another LSTM, whose state is initialised with the representation of the target. Finally, we use the last output vector of the tweet LSTM to predict the stance of the target-tweet pair.

Formally, let ($x_1, \ldots, x_T$) be a sequence of target word vectors, ($x_{T+1}, \ldots, x_N$) be a sequence of tweet word vectors and [h_0 c_0] be a start state of zeros. The two LSTMs map input vectors and a previous state to a next state as follows:

$$\begin{aligned}
[\mathbf{h}_1\ \mathbf{c}_1] &= \mathrm{LSTM}^{\text{target}}(\mathbf{x}_1, \mathbf{h}_0, \mathbf{c}_0) \\
&\ldots \\
[\mathbf{h}_T\ \mathbf{c}_T] &= \mathrm{LSTM}^{\text{target}}(\mathbf{x}_T, \mathbf{h}_{T-1}, \mathbf{c}_{T-1}) \\
[\mathbf{h}_{T+1}\ \mathbf{c}_{T+1}] &= \mathrm{LSTM}^{\text{tweet}}(\mathbf{x}_{T+1}, \mathbf{h}_0, \mathbf{c}_T) \\
&\ldots \\
[\mathbf{h}_N\ \mathbf{c}_N] &= \mathrm{LSTM}^{\text{tweet}}(\mathbf{x}_N, \mathbf{h}_{N-1}, \mathbf{c}_{N-1})
\end{aligned}$$

Finally, the stance of the tweet w.r.t. the target is
classified using a non-linear projection
$c = \tanh(Wh_N)$

where $W € R^{3 \times k}$ is a trainable weight matrix. This effectively allows the second LSTM to read the tweet in a target-specific manner, which is crucial since the stance of the tweet depends on the target (recall the Donald Trump example above).

3. Bi-directional Conditional Encoding

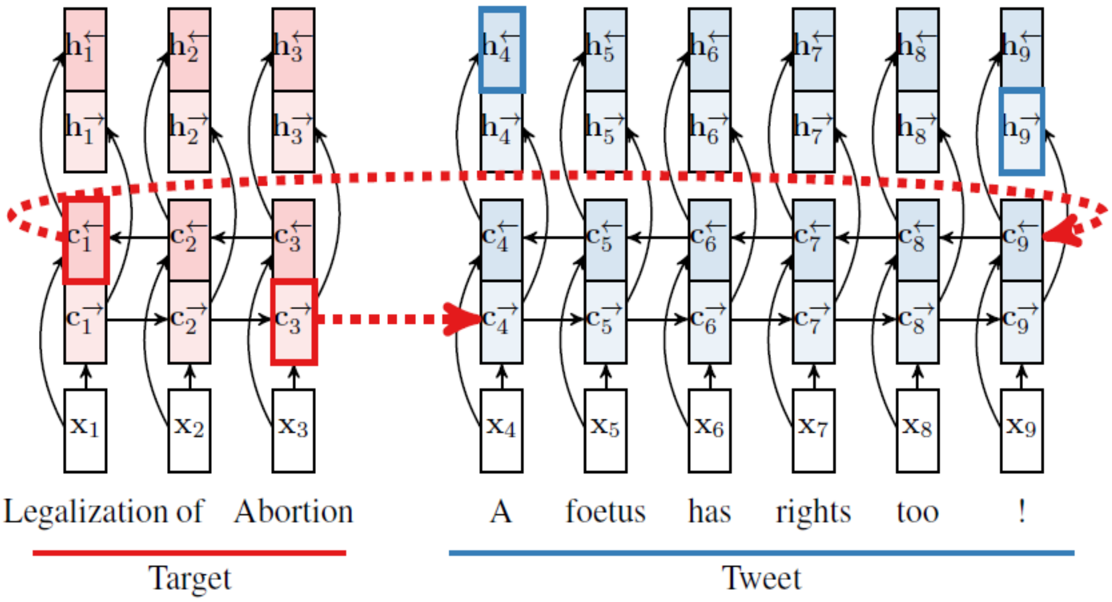

Figure 1: Bidirectional encoding of tweet conditioned on bidirectional encoding of target ([c3, c1]). The stance is predicted using the last forward and reversed output representations ([h9, h4]).

Bidirectional LSTMs have been shown to learn improved representations of sequences by encoding a sequence from left to right and from right to left.

Therefore, we adapt the conditional encoding model from above section to use bidirectional LSTMs, which represent the target and the tweet using two vectors for each of them, one obtained by reading the target and then the tweet left-to-right (as in the conditional LSTM encoding) and one obtained by reading them right-to-left.
To achieve this, we initialise the state of the bidirectional LSTM that reads the tweet by the last state of the forward and reversed encoding of the target (see Figure 1). The bidirectional encoding allows the model to construct target-dependent representations of the tweet such that when a word is considered, both its left- and the right-hand side context are taken into account

PHASE -II:

Objective of CS-529 phase 2 project

To detect the stance for unseen test target as correctly as possible by using bidirectional conditional encoding of Bi-LSTMs representing two vectors for tweets and targets and using word2vec word embedding from Google News.

Evaluate the model after training with different learning rates and observe the corresponding F1-score.

Limitations of proposed model

- The data used in the actual paper had 5,628 instances in total but the dataset we used has 2915 instances (SemEval 2016 Task 6 corpus for Stance Detection on Twitter) in train file and 1957 instances in test file which is almost half of the data they used for training because of which the F1-score wasn’t increasing that much.

- Embeddings used in paper are obtained by unsupervised pre-training with an appropriately trained word2vec model on a corpus of 395,212 unlabelled tweets whereas we used the word2vec model from GoogleNewsvectorsNegative300.bin file .This could be a reason for getting moderate F1-score.

- Data provided was imbalanced so it was more biased towards the class having more data points.

Explanation of intuitions behind objective:

The Bi-LSTM used in the model has bidirectional encoding that allows the model to construct target-dependent representations of the tweet such that when a word is considered, both its left- and the right-hand side context are taken into account.

Word2vec embedding used to get a 300 dimensional embedding for each word to capture the stance of a tweet with respect to a particular target. In order to learn how to combine the stance target with the tweet in a way that generalises to unseen targets, we focus on learning distributed representations and ways to combine them.

We also implemented a model using RoBERTa (pre-trained on twitter corpus) on the same dataset and noticed significant improvement in the F1-score.

We implemented three experiments using small and large datasets and observed corresponding F1-scores for them.

1. MODEL-I: Bidirectional Conditional Encoding on Small Dataset with 2915 Training instances & 1957
2. MODEL-II: Bidirectional Conditional Encoding on Large Dataset with 20K training instances
3. MODEL-III: RoBERTa on Large Dataset with 20K training instances

MODEL-III: RoBERTa on Large Dataset with 20K instances

In this model we have imported AutoTokenizer and AutoModelForSequenceClassification from pretrained libraries for tokenizing and for Stance detection task.

We have passed Cleaned tweets and targets from training and validation data as an input to the model which will in return tokenizes the input sentences and convert them into vectors for passing as an input to the Twitter_Dataset class.

Supporting experimental setup:

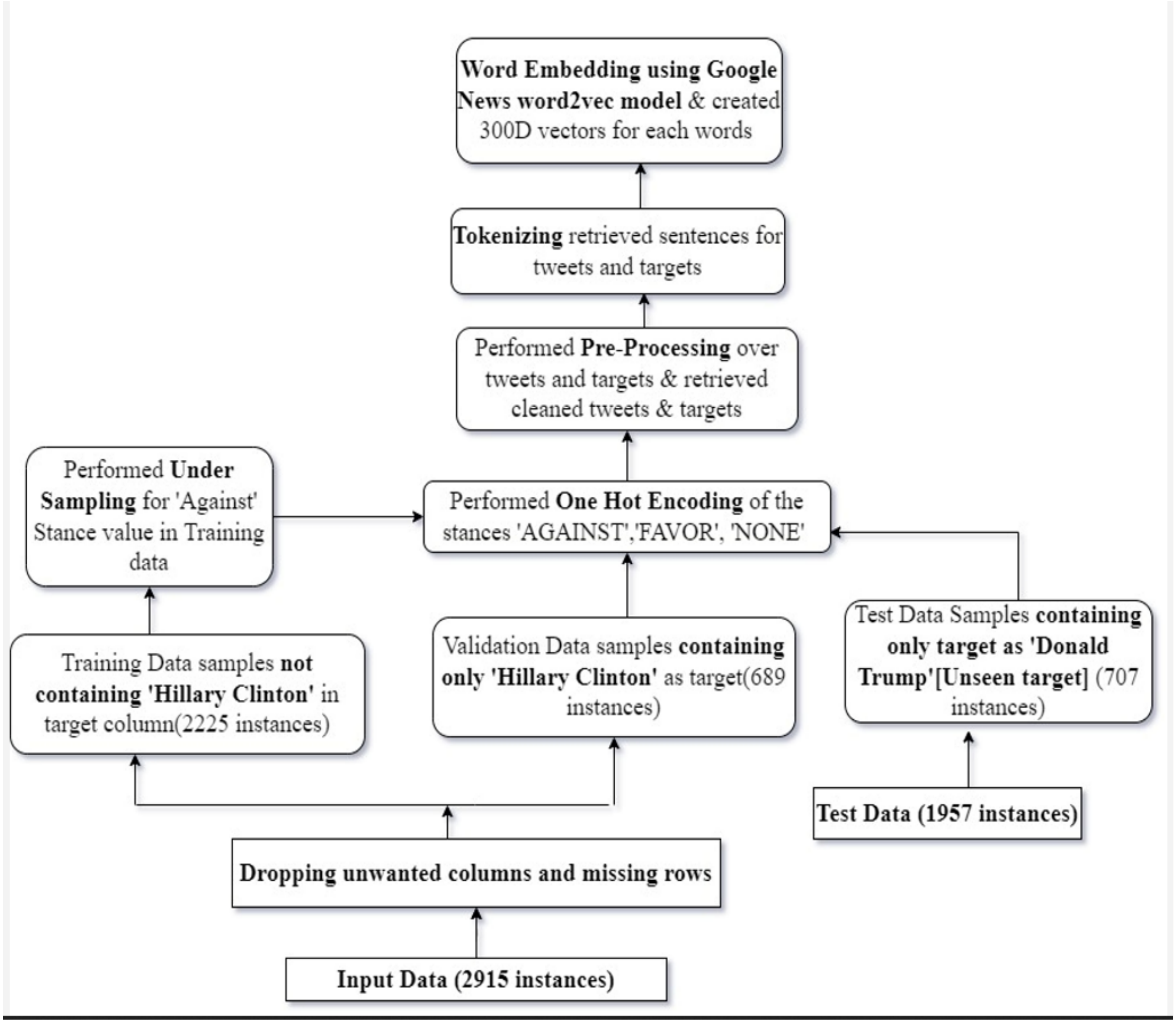

Fig.2 Experimental setup Block Diagram

1. Imported the required libraries with their compatible versions for our stance detection task.
2. Dropped unwanted columns and any missing rows in the dataset.
3. For the Training dataset we took samples containing target other than ‘Hillary Clinton’ and For validation data we took samples containing ‘Hillary Clinton’ as a target. For test dataset, we took only the samples containing ‘Donald Trump’ as target. This was done to test the model on a target which is not seen in the training data.
4. The proportion of data with stance ‘Against’ was more than that with other stances. Hence, we under-sampled the dataset to balance the training dataset.
5. Performed one-hot-encoding for the three stance classes in all train, test and validation data frames.

6. Performed Pre-processing over tweets and targets by removing links representations, symbols, expressions and by removing stop words over all train, test and validation data.
7. Tokenizing all the retrieved sentences from pre-processing of tweets and targets
8. Used word2vec embedding of Google News for creating 300 dimensional vectors for each word.
9. Created two separate Bi-LSTM classes for tweets and targets and performed Bi-conditional encoding. The architecture described in paper was implemented and final results were obtained by using Softmax activation function in the last layer.

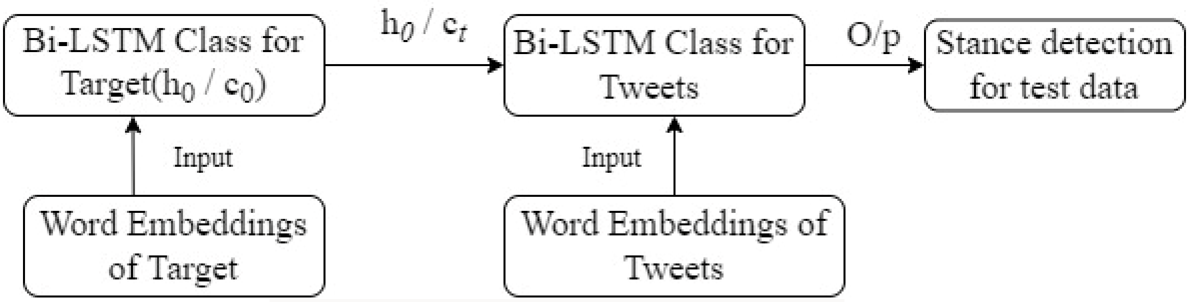

Fig. 3 Basic architecture for Bi-LSTM

10. Trained the entire model end-to-end using Cross Entropy Loss and ADAM optimizer. We report the F1-score on test dataset below.

Observations from experimental results:

Bi-LSTM with conditional encoding on small dataset:

Here we have mentioned weighted F1-scores followed by class wise F1-scores for various batches.

```
21.428571428571423
[21.42857143  0.          0.        ]
34.710743801652896
[34.7107438  0.         0.        ]
93.37647058823529
[94.11764706  0.         20.        ]
77.30061349693253
[77.3006135  0.         0.        ]
87.00564971751412
[87.00564972  0.          0.        ]
19.819819819819816
[19.81981982  0.          0.        ]
40.0
[40.  0.  0.]
The f1_score for the test data is  0.5337740983610372
```

Bi-LSTM with conditional Encoding on large dataset:

```
27.80168067226891
[21.17647059  0.          45.71428571]

28.810126582278485
[28.57142857  0.          30.37974684]

48.82051282051282
[71.79487179  0.           0.         ]

63.9729392914367
[74.45255474  0.          34.14634146]

70.5369807497467
[84.3537415   0.          38.29787234]

19.75996292863763
[16.86746988  0.          28.20512821]

37.10518934081347
[38.70967742  0.          34.7826087 ]
```

RoBERTa Model on Large Dataset:

```
from sklearn.metrics import f1_score

print(100*f1_score(np.array(preds).argmax(1), test_stance.argmax(1), average ='weighted'))

59.44333996023856
```

F1-score will be 0.59

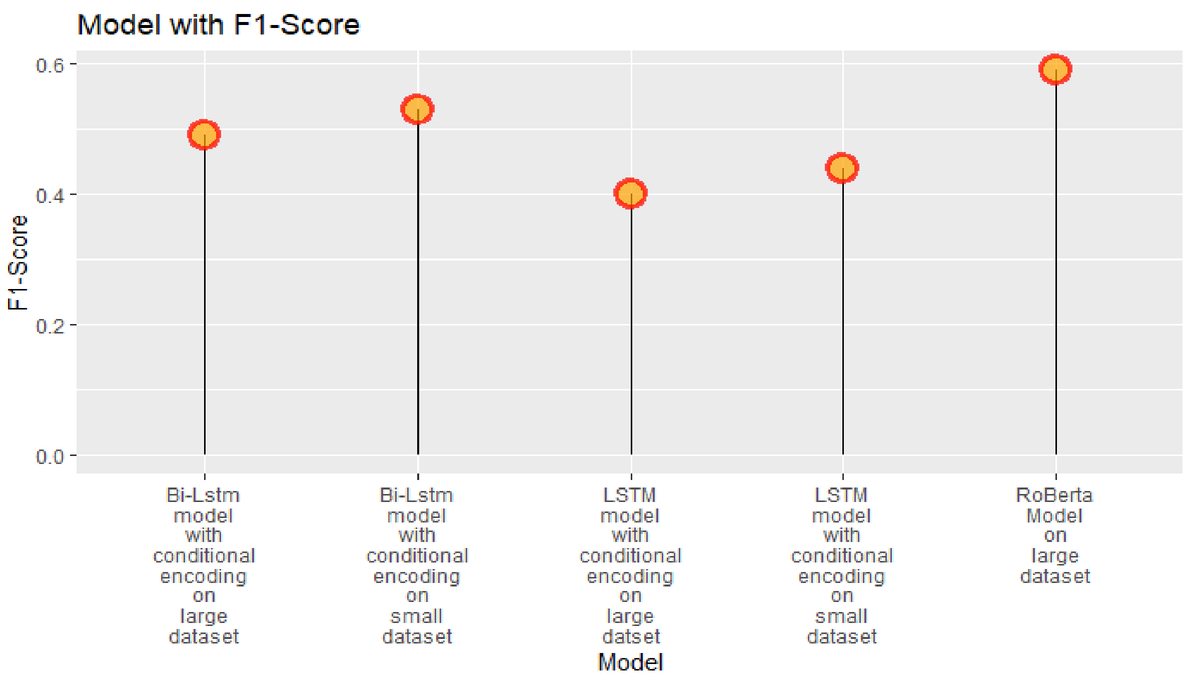

Fig.4 Models implemented with their received F1-Scores

RESULTS:

Results for the unseen target setting show how well conditional encoding is suited for learning target dependent representations of tweets, and crucially, how well such representations generalise to unseen targets. The best performing method on both development and test setups is Bi-LSTM conditional encoding model. However RoBERTa model is giving the best F1-score over large dataset.

Model with F1-score	
RoBerta Model on large data set	.59
Bi-Lstm model with conditional encoding on small data set	.53
Bi-Lstm model with conditional encoding on large data set	.49
LSTM model with coditional encoding on small data set	.44
LSTM model with conditional encoding on large data set	.40

Future work if any :

Instead of passing cell state(c_t) of LSTM_target to initial hidden state directly of LSTM_twitter we can use attention model which calculates weighted sum of all the cell states using LSTM_target .This can further increase the F1-score to some extent.