Rumor Detection

SemEval19

Group 11
Abhijeet Panda, Aniruddha Deshpande,
Darshan Kansagara, Sarvat Ali

Abstract

The task of Rumour Detection here involves determining rumor veracity and the stance taken for each reply in the discussion that follows. Here we focus on stance classification of tweets towards the truthfulness of rumors circulating in Twitter conversations in the context of breaking news. Each conversation is defined by a tweet that initiates the conversation and a set of nested replies to it that form a conversation thread. The task is divided into two Subtasks:
TASK-A. Classifying the replies as to whether they Support, Deny, Query or Comment on the source tweet
TASK-B. Verify the veracity(True, false, unverified) of the source tweet.
As mentioned in our scope document, for this deliverable, the focus was given to Subtask A. This paper describes the various models and methodologies used for the subtask and our future goals for the final deliverable of the project.

1. Stance Classification

Stance Classification of the responses to the source tweet is the first subtask for this project. This involves the categorization of these responses into the following categories:

- **Support**: The author of the response supports the veracity of the rumor they are responding to.
- **Deny**: The author of the response denies the veracity of the rumor they are responding to.
- **Query**: The author of the response asks for additional evidence in relation to the veracity of the rumor they are responding to.
- **Comment**: The author of the response makes their own comment without a clear contribution to assessing the veracity of the rumor they are responding to.

The **dataset** used was provided as a part of **SemEval19** can be found here. This dataset involves both **Twitter** and **Reddit** conversations. We decided to currently focus only on Twitter conversations as the implementation can be easily extended over the Reddit conversations. This is so because both the conversations are structured in a similar tree-like format with source tweets having their veracity labels (ie. True/False), which are joined by an ensuing discussion in which further users support, deny, comment or query (SDCQ) the source text. Following is an illustrated example with a Putin Example:

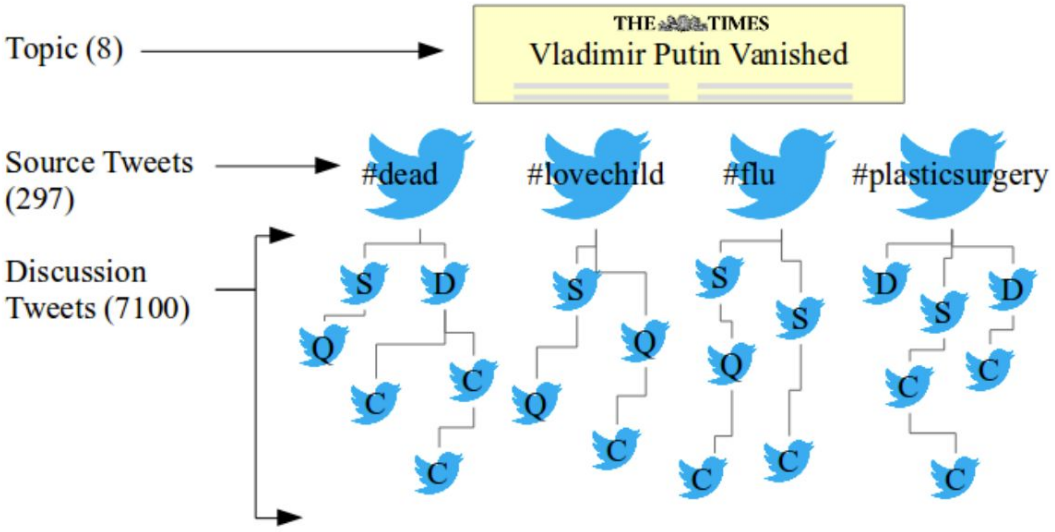

Figure 1: Structure of Twitter Conversations for a given topic.

2. Methodologies tried along with the datasets used

2.1 Dataset Used

- https://figshare.com/articles/RumourEval_2019_data/8845580

2.2 Our Approach

Following steps were carried out as a part of Data Preprocessing:

- Lowercase everything
- Remove user handles & URLs
- Transform hashtags into words

2.2.1 Transformer based model

We used transformer based model BERT (Bidirectional Encoder Representations from Transformers) bidirectional training of Transformer, a popular attention model, to language modeling. Transformer is an attention mechanism that learns contextual relations between words (or sub-words) in a text this approach allows the model to learn the context of a word based on all of its surroundings (left and right of the word). This already Pretrained model on large corpora such as Wikipedia dump and books corpus can be fine-tuned for our Twitter stance classification.

We then used this model for sentence pair-wise classification where each pair corresponds to the source text and its reply and output is whether the given reply supports, deny, query or just comment to given source tweet. We trained the model with various parameters and decided to go forward with the parameters mentioned below in Section 3.1.

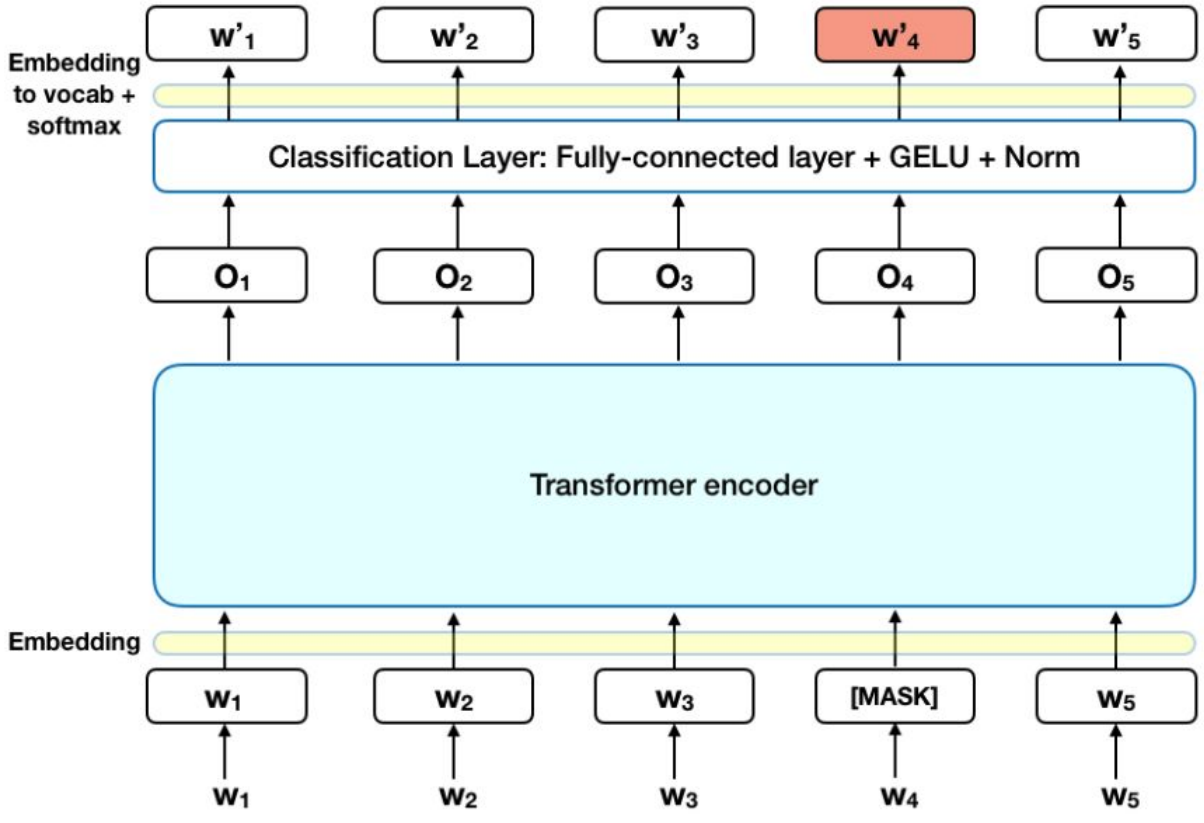

Figure 2: BERT Model Architecture

2.2.2 CNN based model using ELMo embedding

We implemented a CNN-based neural architecture using Elmo embeddings of post text combined with its auxiliary features. Traditional embedding methods such as word2vec or GloVe work independently of the context and always map the same word to the same vector. In contrast, ELMo recent embedding approach is a bidirectional LSTM network that considers the context of the word, that the same word can have different meanings depending on its context.

We represent each text to ELMo embedding, Next, the embedded text is fed into many convolutional layers. Each convolution operation is batch normalized after a ReLU activation. Please refer to Section 3.2 for the hyperparameter values used during training.

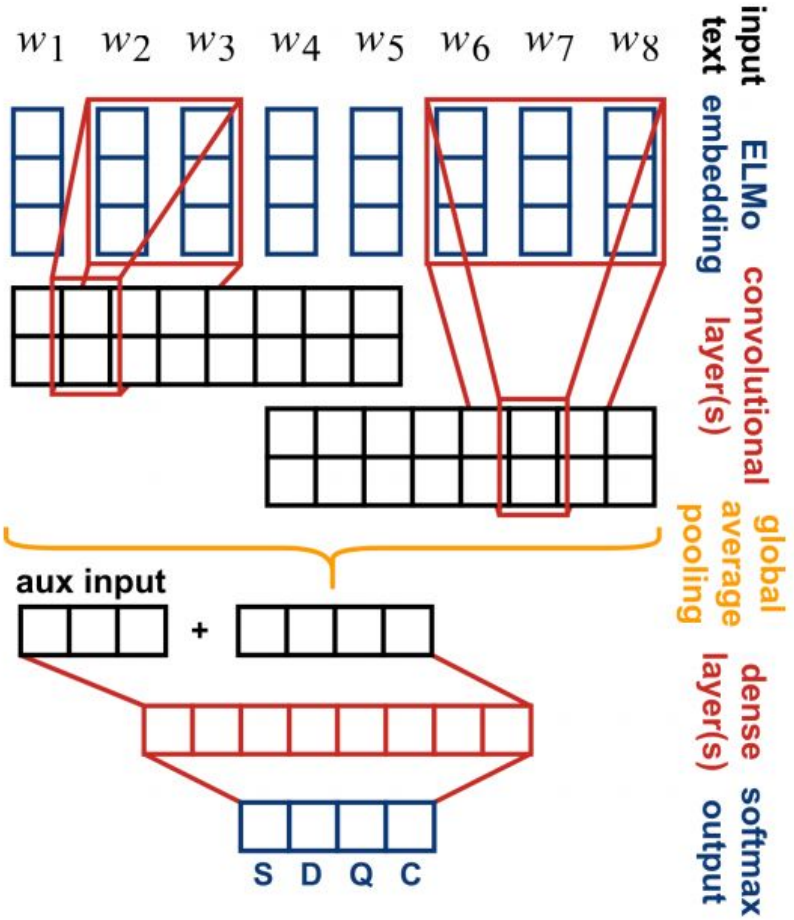

Figure 3: CNN Based architecture with ELMo sentence embeddings.

Please refer to the following section for the scores achieved using the above-used models for Subtask A.

3. Findings from the current implementation

3.1 Transformer based model (BERT)

3.1.1 Hyperparameter Values

Hyperparameter	**Value**
Training Batch Size	32
Learning Rate	5e-5
Number of Epochs	4.0
Warmup Proportion	0.1
Maximum Sequence Length	128
Board Size	19
Drop Out	0.1

3.1.2 Scores Achieved

Class	**Precision**	**Recall**	**F1-score**
Comment	0.78	0.87	0.83
Deny	0.38	0.20	0.26
Query	0.49	0.60	0.54
Support	0.50	0.29	0.37

Task A: SDQC overall Result : Accuracy: 0.72, **F1-Score:** 0.54

3.2 CNN based model

3.2.1 Hyperparameter Values

Hyperparameter	**Value**
Maximum Sentence Length	32
Batch Size	512
Number of Epochs	100
Learning Rate	1e-3
Weight Decay	1e-2
Class Weights	[1,1,1,0.2] for [S,D,Q,C] respectively
Number of Convolutional Layers	1
Kernel Sizes	[2,3]
Number of Channels	64
Number of Dense Layers	3
Number of Hidden Dense Layers	128
Dense Dropout	0.5

3.2.2 Scores Achieved

Class	**Precision (in %)**	**Recall (in %)**	**F1-score (in %)**
Comment	86.2	89.3	87.6
Deny	15.9	12.1	12.7
Query	56.9	33.5	41.7
Support	40.8	37.2	36.9

Task A: SDQC overall Result : Accuracy: 77.7%, **F1-Score:** 44.8%

4. Code link to the baseline implementations

4.1 Github Repository

https://github.com/darshank15/IRE-major-Project---SemEval-Rumour-Detection

4.2 Colab Page

https://colab.research.google.com/drive/1k1h36dV6hITEe8t0shrtTMbZdkwdgzHY

5. Differences compared to the original scope document

In the scope document, we proposed the Neural network model and we moved a step ahead and fine-tuned deep learning-based pre-trained model BERT and CNN based model with ELMO embeddings.

6. Our Future Goals

As specified in the scope document we came with some models for our TASK-A. Now we will be modeling our TASK-B using TASK-A results by **11th November 2019**.
Weekly progress.

WEEK-1	Exploring Pytorch-Transformers library by HuggingFace for additional models for TASK-A consider various twitter specific features to model TASK-A like followers count, favorites-count, etc
WEEK-2	Modeling TASK-B with different classification based models to classify given source tweet as true, false and unverified and return the confidence score (0 means rumor is unverified)
WEEK-3	Evaluate Different approaches for TASK-B and come up with the most appropriate performance parameters.

References

[1] Rani Horev - BERT Explained: State of the art language model for NLP

[2] Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga - RumourEval 2019: Determining Rumour Veracity and Support for Rumours

[3] Thilina Rajapakse - A Simple Guide On Using BERT for Binary Text Classification

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[5] Martin Fajcik, Lukáš Burget, Pavel Smrz - SemEval-2019 Task 7: Determining the Rumour Stance with Pre-Trained Deep Bidirectional Transformers

[6] Elena Kochkina, Maria Liakata, Isabelle Augenstein - Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM

[7] Ruoyao Yang, Wanying Xie, Chunhua Liu, Dong Yu - BLCU NLP at SemEval-2019 Task 7: An Inference Chain-based GPT Model for Rumour Evaluation