Columbia at SemEval-2019 Task 7: Multi-task Learning for Stance Classification and Rumour Verification

Zhuoran Liu* **Shivali Goel*** **Mukund Yelahanka Raghuprasad**
Department of Computer Science, Columbia University
{zhuoran.liu, shivali.goel, my2541}@columbia.edu

Smaranda Muresan
Data Science Institute, Columbia University
sm761@columbia.edu

Abstract

The paper presents Columbia team's participation in the SemEval 2019 Shared Task 7: RumourEval 2019. Detecting rumour on social networks has been a focus of research in recent years. Previous work suffered from data sparsity, which potentially limited the application of more sophisticated neural architecture to this task. We mitigate this problem by proposing a multi-task learning approach together with language model fine-tuning. Our attention-based model allows different tasks to leverage different level of information. Our system ranked 6th overall with an F1-score of 36.25 on stance classification and F1 of 22.44 on rumour verification.

1 Introduction

The ubiquity of social media is allowing unverified news and rumours to spread easily. Efforts have been made to explore automated methods for rumour detection and verification (Derczynski et al., 2017; Zubiaga et al., 2016, 2018), and has shown promising potential to tackle this issue at scale.

RumourEval 2019 Shared Task 7 tackles the problem of predicting the veracity of rumours and stance of replies. It consists of two subtasks: task A (SDQC), in which stance (support, deny, querying, comment) of responses to a rumourous statement are predicted, and task B (verification), in which the statement's veracity is to be predicted. Size of training data provided for Task A is 5,217 and for Task B is 327.

In this paper, we proposed several methods to alleviate data sparsity and unleash the power of sophisticated neural models:

1. *Jointly learning to perform rumour verification and stance detection.* Training a neural network on limited amount of data for a single task is hard, especially in a sentence classification task. This is because of the weak supervision signal caused by the information asymmetry between the source text and the target labels. With supervision signal from multiple tasks, a neural network can exploit information in the training data more thoroughly.

2. *Using self-attention.* To predict the stance of a post, we want to selectively pay attention to some other posts that are relevant to this post. We use a QKV-style attention (Query, Key, Value) (Vaswani et al., 2017) to summarize the post context into a single vector (where in practice one attention head is usually enough). In addition, we use representations at different levels for different tasks.

3. *Using language model fine-tuning for stance classification.* We use the Universal Language Model Fine-tuning (ULMFiT) (Ruder and Howard, 2018) to improve our stance classifier. We begin with a generic language model trained on the Wikitext 103 dataset (Merity et al., 2016). This dataset consists of a large collection of pre-processed English Wikipedia articles. This enables the language model to properly model the general properties of language. Next, we fine-tune this language model on task specific data: RumourEval2019 dataset. Finally, a classification layer is added and the model is initialized with parameters from the fined-tuned language model.

Our system, which relies on these three key factors, are now publicly available.[2]

* Equal contribution.

[2] Github repository: https://github.com/joelau94/rumour2019-experiments

2 Related Work

Rumour Detection. Recently there has been a growing interest on developing methods for the task of rumour detection (Zubiaga et al., 2018), including a shared task in 2017 (Derczynski et al., 2017), which established a strong baseline for stance classification — task A(Kochkina et al., 2017), while (Enayet and El-Beltagy, 2017) established the same for veracity — task B. Dungs et al. (2018) discuss how stance information can facilitate veracity classification, while (Zubiaga et al., 2017) explore the use of contextual information for rumour detection. These results show that stance information and context information are important for rumour verification.

Multi-task Learning. Text classification tasks invariably suffer from the weak supervision signal due to loss of information in projecting text to task labels. There has been a growing number of works that explore multi-task learning for text classification (Zhang et al., 2017; Xiao et al., 2018). For the task of rumour detection specifically, there were attempts in jointly train for stance classification and rumour verification (Kochkina et al., 2018). Our muti-task approach uses a different, more advanced sentence embedding approach and uses the same LSTM for both tasks but with hidden states from different levels, which can be considered as different level representations of sentences. Empirically we found that higher levels of representation performs better for stance classification, while lower levels are better for veracity classification.

Transfer Learning with pre-trained Language Models. To alleviate the problem of data scarcity, researchers have proposed various approaches for pre-training language models on large-scale monolingual corpora, such as ELMo, ULMFiT, BERT, GPT, and have shown their effectiveness on several NLP tasks (Peters et al., 2018; Ruder and Howard, 2018; Devlin et al., 2018; Radford, 2018). In our work we use ULMFiT (Ruder and Howard, 2018) for stance classification.

3 System Description

We propose two system configurations:

1. *System1:* A joint-learning for task A and task B without using language model fine-tuning.
2. *System2:* Language model fine-tuning for task A.

3.1 System1: Joint Training for Stance Classification and Rumour Verification

We formulate the joint learning of Task A and B as follows: Given a branch of conversation $\mathbf{X}$ containing n posts

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n),$$

where each post $\mathbf{x}_k$ is a sequence of m_k words:

$$\mathbf{x}_k = (x_{k,1}, x_{k,2}, \cdots, x_{k,m_k}).$$

The goal is to build two neural probabilistic models $p(\mathbf{y}_{\text{stance}}|\mathbf{X}; \theta, \phi_{\text{stance}})$ and $p(y_{\text{veracity}}|\mathbf{X}; \theta, \phi_{\text{veracity}})$, where θ is the shared parameters, ϕs are parameters unique to each task, $\mathbf{y}_{\text{stance}} = (y_1, y_2, \cdots, y_n)$ are stance labels, and y_{veracity} is the veracity label.

To estimate θ and ϕs, we perform maximum likelihood estimation (MLE) over the training dataset $\mathcal{D} = \{(\mathbf{X}_d, \mathbf{y}_d)\}_{d=1}^{N}$, with optimization objectives being negative log-likelihoods:

$$\begin{aligned}\mathcal{J}_{\text{stance}} &= -\sum_d \log p(\mathbf{y}_{\text{stance}}|\mathbf{X}; \theta, \phi_{\text{stance}}) \\ &= -\sum_d \sum_i \log p(y_i)_{\text{stance}}|\mathbf{X}; \theta, \phi_{\text{stance}}) \\ \mathcal{J}_{\text{veracity}} &= -\sum_d \log p(y_{\text{veracity}}|\mathbf{X}; \theta, \phi_{\text{veracity}})\end{aligned}$$

Rumour verification and stance classification are highly-related tasks that can potentially provide useful information for each other. Therefore we integrate the two tasks for joint training, allowing more accurate estimation of the shared part of parameters θ.

To find an appropriate balance between the supervision signals from the two tasks, we introduce a tunable hyper-parameter λ. We then rewrite our objective function as follows:

$$\mathcal{J} = \lambda \cdot \mathcal{J}_{\text{stance}} + (1 - \lambda) \cdot \mathcal{J}_{\text{veracity}}$$

We designed an effective neural network to model $p(\mathbf{y}_{\text{stance}}|\mathbf{X}; \theta, \phi_{\text{stance}})$ and $p(y_{\text{veracity}}|\mathbf{X}; \theta, \phi_{\text{veracity}})$, which provides latent structures to capture subtleties of conversations. This model architecture is described below.

Neural Network Architecture

Inspired by the idea of BranchLSTM (Kochkina et al., 2017), we propose a model based on a single branch from the conversation tree. Our model is different from BranchLSTM (Kochkina et al., 2017) in the following ways:

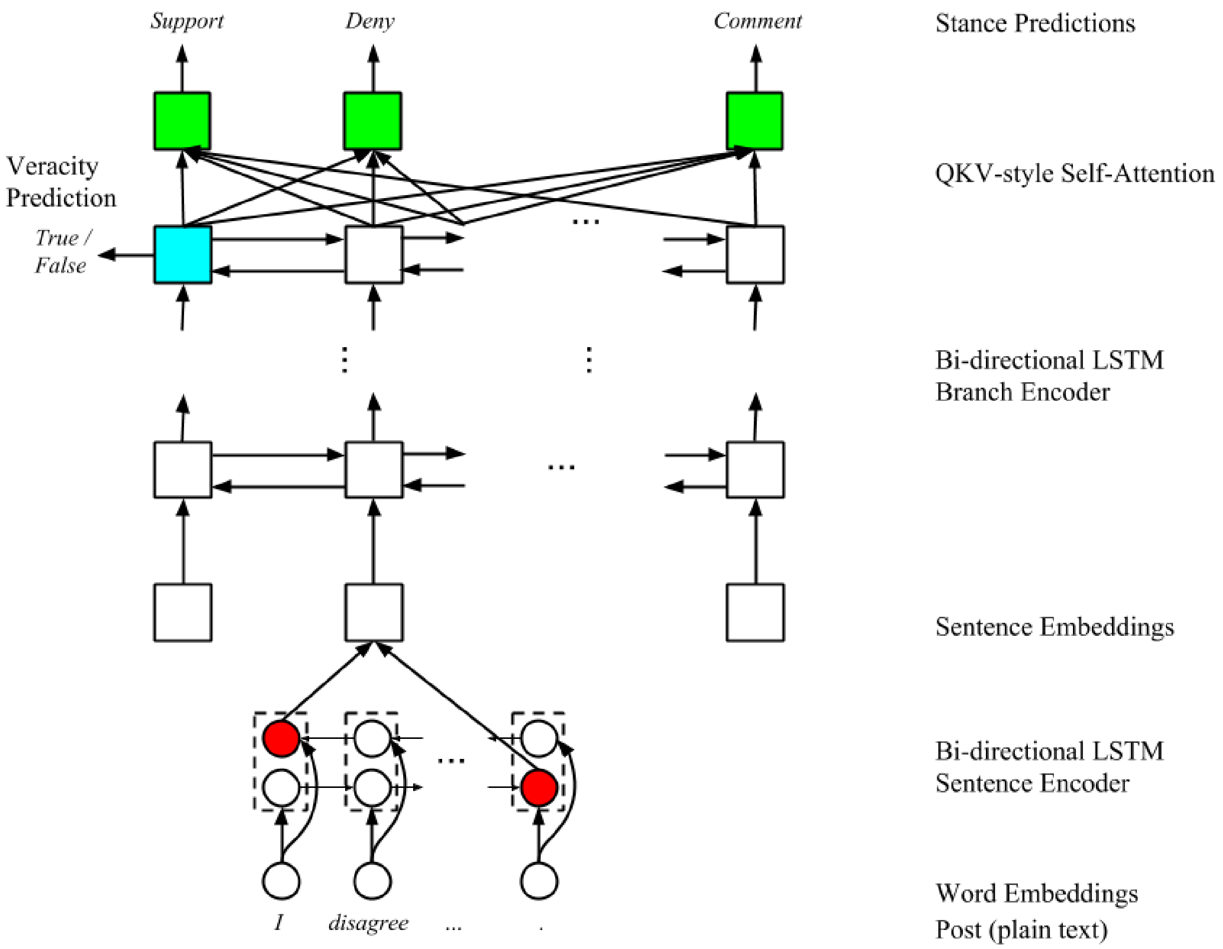

Figure 1: Model Architecture.

1. The sentence vector representation (sentence embedding) is generated with a bi-directional LSTM, as compared to simply taking the average of word vectors in BranchLSTM. This allows sentence embeddings to selectively encode important words and capture long-distance dependency in the sentence.

2. We apply a Transformer-style attention (Vaswani et al., 2017) on top of branch-level LSTM. This enables the most important information to flow in when trying to decide the stance of a post.

3. Rumour verification is incorporated as a task being jointly learnt together with stance classification, yet exploiting information at a different level from stance classification. In practice, hidden states at different levels of LSTM is being used for different tasks.

Figure 1 shows our overall model architecture, which we describe in more detail below.

Word Embeddings. The word embedding space is adjustable in our model. We initialize the word embedding matrix with pre-trained GloVe embeddings (Pennington et al., 2014). While we fix most word embedding vectors, we also keep some of the most frequent word embeddings trainable, allowing the word embedding space to adjust itself.

Sentence Embeddings. We consider each post as a sentence and we encode it with a bi-directional LSTM. We then take the last hidden state of the forward LSTM and first hidden state of the backward LSTM and concatenate them. The resulting vector can be considered as a dynamically generated sentence embedding.

Stacked Branch Encoder. To capture the interaction between posts in a branch of conversation, we use a stacked Bi-LSTM to encode the sentence embeddings obtained from previous steps. This results in a higher level representation of each post, which is fully aware of the conversation context. The higher the level in the LSTM stack, the more the representation is aware of context.

Attention. To predict the stance of a post, we want to selectively pay attention to some other posts that are relevant to this post. We use a QKV-style of attention (Vaswani et al., 2017) to summarize the post context into a single vector (where in practice one attention head is usually enough).

Stance Classifier. We first send the highest-level representation of posts to a QKV-style self-attention, which produces an attention-weighted context vector for each post in a branch. We then concatenate each post's representation with its corresponding context vector, and feed it through an MLP followed by a softmax for stance classification. During our experiment, we found that one attention head is good enough and is better than using multi-head attention.

Veracity Classifier. We take the representation of the original post (which is the first post in each conversation) from some intermediate layer, and feed it through an MLP followed by a softmax for veracity classification. This design corresponds to the intuition that when judging the authenticity of a post, the model should focus more on the post itself and less on how people judge it.

All hyperparameters can be found in our code with default settings.

3.2 System2: LM fine-tuning for Stance Classification

We also tried improving our stance classifier by using the Universal Language Model Fine-tuning (ULMFiT). After training a generic language model trained on Wikitext 103 dataset, we fine-tuned the LM on RumourEval2019 dataset.

Pre-processing was inspired by BranchLSTM system (Kochkina et al., 2017). Tweets along a particular branch were concatenated starting from the source tweet till the target node and considered as one training instance. The SDQC label of the last node concatenated was considered to be the label of the training instance. Here, replies are being referred to as nodes. For instance, source + reply 1 + reply 2, label of reply 2 was one training instance.

Finally, we used a BiLSTM max pooling network which was presented in (Conneau et al., 2017) and is shown in Figure 3. This model was initialized with parameters from the fined tuned language model. In this architecture, the representation generated by the BiLSTM was max pooled, i.e. the maximum value over each dimension was selected to form a fixed-size vector and was followed by softmax for stance classification.

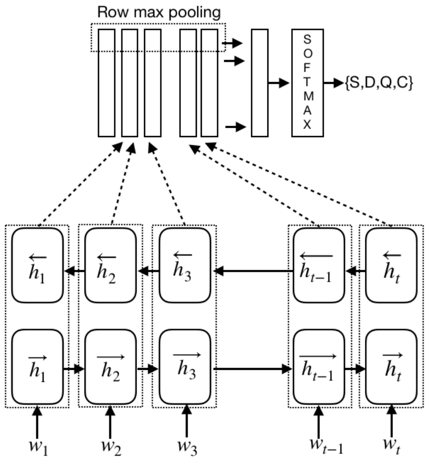

Figure 2: Max pooling in System2

4 Results and Analysis

For System1, we achieved an F1-score of 22.44 on task B and an F1-score of 34.04 on task A. For System2, we achieved F1-score of 36.25 on task A (System2 is only applied to task A). Performance of System1 on task A is slightly lower than the performance of System2, because we only treat task A as an auxiliary task for task B and did not apply ULMFiT to task A. Our final submission consisted of using System 2 for task A and System1 for task B. Our final submission ranked 6th in the final leaderboard.

	Verif	RMSE	SDQC
System 1	0.2244	0.8623	0.3404
System 2			0.3625
Final Submission	0.2244	0.8623	0.3625

Table 1: Performance of two systems on test set.

Unbalanced Class Labels. The model in System1 suffers heavily from an unbalanced class problem. From Table 2, we can see that the model is not giving any predictions of D (Deny) and Q (Query), which is why even though it has high accuracy (83%+), its F1 is lower than that of System2.

	S	D	Q	C
System 1	81	0	0	1746
System 2	62	16	84	1665

Table 2: Predicted class frequencies of SDQC classification on test data.

This problem is mitigated a little bit in System2, as we witnessed a few examples of D and Q predictions. This could potentially be because of the general knowledge gained by pre-training on large-scale Wikipedia text. Even then, D and Q classes are rare in the model predictions.

Instability in Training. System1 shows a perturbing training loss after it decrease to a certain level. After a certain point, the F1 score and accuracy on development set begins decreasing. One explanation is that the size of training data is too small and noise in the data negatively impacts the model.

5 Conclusion

In this work, we present the Columbia Team's system submission for the RumourEval 2019 shared task. We tackle the issue of data sparsity by multitask learning which fully utilizes the training data. In addition, we also apply pre-training techniques such as ULMFiT which was effective in improving results on task A.

References

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In *SemEval@ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sebastian Dungs, Ahmet Aker, Norbert Fuhr, and Kalina Bontcheva. 2018. Can rumour stance alone predict veracity? In *COLING*.

Omar Enayet and Samhaa R. El-Beltagy. 2017. Niletmrg at semeval-2017 task 8: Determining rumour and veracity support for rumours on twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 470–474, Vancouver, Canada. Association for Computational Linguistics.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm. In *SemEval@ACL*.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. In *COLING*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Alec Radford. 2018. Improving language understanding by generative pre-training. In *Preprint*.

Sebastian Ruder and Jeremy Howard. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Liqiang Xiao, Honglun Zhang, and Wenqing Chen. 2018. Gated multi-task network for text classification. In *NAACL-HLT*.

Honglun Zhang, Liqiang Xiao, Yongkun Wang, and Yaohui Jin. 2017. A generalized recurrent neural architecture for text classification with multi-task learning. In *IJCAI*.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Comput. Surv.*, 51:32:1–32:36.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In *COLING*.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *SocInfo*.