The Great Rumour Showdown: Tweets vs. Threads

Marija Anđelić, Antonio Lukić, Marko Turina

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
Marija.Andelic,Antonio.Lukic,Marko.Turina@fer.hr

Abstract

The internet is a widespread platform where anyone can share their thoughts and opinions, often leading to the spread of unverified information and false rumours. Social media platforms like Twitter and Reddit are particularly susceptible to this phenomenon, where users frequently post and react to content without verifying its accuracy. This proliferation of misinformation presents significant challenges in detecting and managing false rumours. In this paper, we investigate the impact of learning with heterogeneous datasets on final scores in rumour detection tasks. Additionally, we examine how the utilization of only replies versus incorporating both source texts and replies affects the categorization of replies in social media rumour analysis.

1. Introduction

The emergence of the Internet and the widespread adoption of social media platforms have revolutionized the way information is shared and consumed. With the ability for anyone to publish content online, the task of detecting and managing rumors and unverified information becomes increasingly challenging. This proliferation of misinformation poses significant challenges in discerning truth from falsehood, particularly on platforms like Twitter and Reddit where user-generated content can quickly go viral.

The structure and format of social media posts vary significantly across platforms, resulting in the generation of heterogeneous datasets sourced from diverse platforms. Twitter, with its character limit, enforces concise and often fragmented discussions, while Reddit allows longer-form content and threaded conversations. As a result, rumour detection and verification tasks are complicated further by the challenges posed by these heterogeneous datasets.

In this paper, we investigate how learning with datasets collected from diverse sources, such as Twitter and Reddit, affects the final scores in rumour detection tasks. By analyzing the performance of models trained on individual platforms versus combined datasets, we aim to gather insights into the generalization capabilities and limitations of rumour detection algorithms across different social media platforms. Furthermore, we examine the effect of using only replies versus including both source texts and replies in the categorization of responses to rumorous posts. Our goal is to clarify the importance of contextual information by comparing how models perform under both of these conditions.

For this purpose, we utilize the dataset provided for The SemEval-2019 Task 7 (Gorrell et al., 2018), focusing specifically on the subtask A. This subtask entails categorizing replies to rumorous posts into four categories: *support*, *denial*, *queries*, or *comments*.

2. Related Work

A growing interest in rumour detection and verification has led to numerous studies tackling this issue. In addressing the challenges of sparse and disproportionate labels in rumour detection, Yang et al. (2019) proposed an inference chain-based system that leverages conversation structure and rich intrinsic features, achieving top performance in subtask A. Their method addresses class imbalance through a two-step classification process and expands training data for minority classes using similar external datasets. Khandelwal (2020) introduced a multi-task learning framework that jointly predicts rumor stance and veracity on the dataset. Their approach models the multi-turn conversation, ensuring that each post in the thread is aware of its neighboring posts. The bottom part of their framework classifies the stance for each post, demonstrating superior performance over previous approaches and generalizing effectively across various social media platforms.

Liu et al. (2024) proposed a new detection model that learns the joint representations of user correlation and information propagation to detect rumours on social media. They utilized graph neural networks (GNNs) to capture complex relationships and interactions between users and posts. Their model outperformed the state-of-the-art rumour detection models. Shehata et al. (2024) examined the use of conversational prompt-engineering-based large language models (LLMs) to address digital misinformation and rumour detection. Their findings indicated that models GPT-4 and GPT-3.5-turbo can effectively predict the veracity of rumours, often outperforming previous models in veracity prediction. However, for stance classification, the improvements were less significant compared to those achieved by fine-tuned methods.

While some previous works have expanded the dataset by incorporating similar datasets (as in (Yang et al., 2019)) or by employing other methods, this paper concentrates exclusively on the given dataset and explores what can be accomplished using only its data.

3. Dataset

The dataset used in this paper was created as part of the SemEval-2019 workshop for Task 7. It focuses on detecting and resolving rumors on social media, encompassing tasks like rumor detection, stance classification, and veracity prediction. Source posts introduce a rumor, which can be true, false, or unverified. These posts are followed by a tree-

a) Twitter example:

u1: Witnesses say the Charlie Hebdo gunmen identified themselves as members of al-Qaida: http://t.co/WSEe7PGIdY **[support]**

u2: “@u1: Witnesses-Charlie Hebdo gunmen identified themselves as members of al-Qaida:” If true, terrorist groups competing and escalating? **[query]**

u3: @u1 @u2 Identified themselves ? You having a laugh, so how come they didn’t get shot too ? :/ France **[query]**

u3: @u1 Cant be al-Qaida the President told us they are in retreat **[deny]**

b) Reddit example:

u1: ”Cancer is a fungus” - this idea from the 60s is apparently receiving new attention. Please advise. **[comment]**

u2: To me, it seems ridiculously far-fetched, but I’m trying to be subjective. (Which is hard, considering that Mr. Icke treats the lizard-men myth, among others, with the same respect as this.) **[deny]**

u3: ”but I’m trying to be subjective.” Do you mean objective? Just wow, there is so much wrong with this article I don’t even know where to begin. **[query]**

u2: Oops, of course. Apparently big words aren’t so much my thing. **[comment]**

Figure 1: Example a) features tweet tree structure with labels. Example b) features Reddit thread structure with labels.

shaped discussion where users support, denial, question, or comment on the rumor mentioned in the source text. Every source tweet contains its replies, i.e. reactions. Reddit data is organized in the form of threads, which include a source post along with its replies. Figure 1 illustrates the structure of examples from both Reddit and Twitter datasets.

The dataset of Twitter posts is significantly larger than the Reddit dataset, with 4519 examples in the training set and 1066 examples in the test set, compared to the Reddit dataset which contains 686 examples in the training set and 690 examples in the test set. The Reddit dataset is not only smaller but also highly imbalanced, with 87.5% of examples in the training set and 91.2% of examples in the test set belonging to the class *comment*. In contrast, the Twitter dataset is somewhat more balanced, with 64.3% of examples in the training set and 72.3% of examples in the test set being classified as *comment*. The validation set consists of 429 examples, with 92.3% of examples being classified as *comment*.

4. Models and Hyperparameter Optimization

In this section we explain which models we use, how we represent words and how we optimise our hyperparameters. For this paper, the models and experiments were conducted using the PyTorch library (Paszke et al., 2019). To address our questions and assess how heterogeneous datasets affect the final model, we aim to train a model capable of achieving adequate results. For this purpose, we employ several models and techniques.

4.1. Baseline Model

As a baseline, we use a Logistic Regression model (LR). This model leverages TF-IDF (Term Frequency-Inverse Document Frequency) to represent text data as vectors. The TF-IDF representation helps in highlighting the importance of terms relative to the document and the entire corpus, providing a solid foundation for comparison with more complex models.

4.2. biLSTM and textCNN Models

We train biLSTM (Huang et al., 2015) and textCNN (Kim, 2014) models.

- **biLSTM Model**: This model captures the sequential nature of text data and manages to capture long-range contextual dependencies.
- **textCNN Model**: This model is effective in identifying structures in short texts (posts) and is expected to capture local patterns in the data.

Comparing the performance of these models allows us to evaluate how different architectures respond to different datasets.

4.3. BERT

We fine-tune the BERT (Devlin et al., 2019) base model (uncased), leveraging its pre-trained deep bidirectional transformers. Fine-tuning BERT allows for efficient adaptation to specific domain data, enhancing model performance.

4.4. Embeddings

To represent words in our textCNN and biLSTM models we use GloVe embeddings (Pennington et al., 2014). GloVe provides pre-defined dense vectors for over 6 billion English words. Unlike one-hot encoding, where each word has its dimension, GloVe embeddings offer the opportunity to represent similar words with similar vectors in a lower-dimensional space.

4.5. Hyperparameter Optimization

To determine the optimal hyperparameters, we perform grid-search optimization across different configurations on the validation dataset. We test various batch sizes {16, 32, 64} as well as learning rates {2e-5, 2e-4, 2e-3} alongside different numbers of epochs {2, 3, 4, 5}. This approach allows us to systematically evaluate the performance of the model under different settings and select the combination of hyperparameters that yields the best results for our task. During training, we use AdamW optimizer (Loshchilov and Hutter, 2017).

5. Experimental Results

In this section, we explain the setup of our experiments and discuss our results. Macro F1 score was used as an evaluation metric since it was the most common evaluation metric used in related works.

5.1. Setup

We fine-tune our models using various configurations, each applied to a different dataset setup. Our work could be divided into 4 different experimental setups:

1. Fine-tune the model exclusively on either the Twitter or Reddit dataset
2. Fine-tune the model on both datasets at once
3. Fine-tune the model on both datasets, with the minority classes sampled to match the number of the majority class
4. Fine-tune the model on one dataset and evaluate it on the other dataset

For each approach, we try two methods. In the first method, the model classifies replies solely based on their text. In the second method, additional context from the source is included when classifying the replies. Initially, the model attempts to classify the source post, followed by all reply posts. The input to the model is then as follows:

- without context: `reply_text`
- with context: `reply_text [LABEL] predicted_label_source [SEP] source_text`

5.2. Fine-tuning on One Dataset

Table 1 presents the macro F1 score performance of various models trained on a single dataset, with and without context. Both BiLSTM and textCNN achieved better results than our baseline LR model for both datasets. BERT model trained on the Twitter dataset achieved the best result of them all, but BERT model trained on the Reddit dataset achieved the same results as the LR model. The results demonstrate that model performance varies significantly between the Twitter and Reddit datasets. Models generally perform better on the Twitter dataset compared to the Reddit dataset. This is possibly due to the higher imbalance of the Reddit dataset. There is a decrease in results when using context compared to using only text of replies.

Model	Context	No Context
LR (Twitter)	0.282	-
LR (Reddit)	0.241	-
BiLSTM (Twitter)	0.234	0.243
BiLSTM (Reddit)	0.243	0.286
textCNN (Twitter)	0.341	0.398
textCNN (Reddit)	0.305	**0.323**
BERT (Twitter)	0.402	**0.419**
BERT (Reddit)	0.241	0.241

Table 1: Macro F1 score performance models trained on one dataset

5.3. Fine-tuning on Both Datasets

Table 2 shows the results of models trained on both datasets combined. In this case BERT model achieved by far the best results. This could be due to combined datasets giving out more information than the individual datasets by themselves. Once again, the results of data using context are somewhat lower than of the data without context.

Model	Context	No Context
LR	0.282	-
BiLSTM (both)	0.212	0.221
textCNN (both)	0.232	0.204
BERT (both)	0.403	**0.444**

Table 2: Macro F1 score performance models trained on both datasets

5.4. Fine-tuning on Balanced Datasets

The combined dataset has 3402 instances of comment class. To construct the balanced dataset, additional instances were sampled from other three classes to match the number of instances in the comment class, resulting in each class comprising 3402 instances in the training dataset. Table 3 shows the results of models trained on synthetically balanced datasets. Balancing the datasets did not significantly improve the performance of the BERT model. However, it did enhance the results for the BiLSTM and textCNN models, which achieved better outcomes compared to using the unbalanced combined datasets. Once more the results of data without context are better than the results of data with context of the source post.

5.5. Training on One Dataset and Evaluating on the Other

Table 4 shows the results of BERT models trained on one dataset and tested on the other. The model achieved better results when trained on the Twitter dataset and evaluated on the Reddit dataset than vice versa. This could indicate that

Model	Context	No Context
LR (both balanced)	0.271	-
BiLSTM (both balanced)	0.281	0.292
textCNN (both balanced)	0.252	0.294
BERT (both balanced)	0.391	0.414

Table 3: Macro F1 score performance models trained on both balanced datasets

when merging the datasets, Twitter data contributes more than Reddit data. In the case of BERT trained on the Twitter dataset, we could see a slight improvement when using the context of the source post.

Model	Context	No Context
BERT (train on Reddit)	0.226	0.227
BERT (train on Twitter)	0.332	0.331

Table 4: Macro F1 score performance models trained on one dataset and evaluated on the other

5.6. Analysis

Overall, the models performed worse when the context of the source post was provided compared to when it was not. The source text might contain information that is not directly relevant to the stance of the reply, introducing noise and making it harder for the model to focus on the key features that determine the stance. The architecture of the model and the nature of the training data also play a crucial role. If the model is not well-suited to handle longer inputs or if the training data does not consistently benefit from added context, performance can drop. Lastly, training models independently on a single dataset did not show much difference compared to training on a combined dataset except for BERT.

6. Conclusion

In this paper, we have presented the challenging problem of verifying rumors in social media posts. We utilized the dataset provided for The SemEval-2019 Task 7, focusing on categorizing replies to rumor posts into support, denial, queries, or comments. We experimented with various models including logistic regression, BiLSTM, textCNN, and BERT, and compared their performances. We investigated the effect of using only replies versus incorporating both source texts and replies in the categorization of responses. Our study found that while transformer-based models like BERT generally outperformed traditional models like logistic regression, the performance varied significantly depending on the dataset and the inclusion of contextual information. Models trained on Twitter data generally performed better, likely due to the higher imbalance and smaller size of the Reddit dataset.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. Minneapolis, Minnesota, June. Association for Computational Linguistics.

Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. Rumoureval 2019: Determining rumour veracity and support for rumours.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging.

Anant Khandelwal. 2020. Fine-tune longformer for jointly predicting rumor stance and veracity.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Tianrui Liu, Qi Cai, Changxin Xu, Bo Hong, Fanghao Ni, Yuxin Qiao, and Tsungwei Yang. 2024. Rumor detection with a novel graph neural network approach.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. Doha, Qatar, October. Association for Computational Linguistics.

Dahlia Shehata, Robin Cohen, and Charles Clarke. 2024. Rumour evaluation with very large language models.

Ruoyao Yang, Wanying Xie, Chunhua Liu, and Dong Yu. 2019. BLCU_NLP at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation. Minneapolis, Minnesota, USA. Association for Computational Linguistics.