Detecting Rumor Veracity with Only Textual Information by Double-Channel Structure

Anonymous ACL submission

Abstract

We develop a double-channel classifier to detect the veracity of social media rumors, relying only on the most basic textual information. Our model first assigns each thread into a "certain" or "uncertain" category. Since authors with a proprietary source of information are likely to post threads with a certain textual tone, we apply lie detection algorithms to certain texts. In contrast, as uncertain threads are arbitrary, we examine whether the replies are in accordance with the threads instead of applying the lie detection algorithms. This approach yields a macro-F1 score of 0.4027, outperforming all the baseline models and the second-place winner of SemEval 2019 Task 7. Further, we show that dividing the sample into two subgroups significantly improves the classification accuracy, reinforcing our claim that applying appropriate classifiers is crucial in rumor veracity detection.

1 Introduction

This study develops a model that automatically detects the veracity (true, false, or unverifiable) of a rumor in social media such as Reddit and Twitter. We employ a double-channel model to improve the accuracy of veracity detection while using only the textual information set available at the time of rumor origination. We train and evaluate the performance of our model with the dataset released in SemEval 2019 Task 7 (Gorrell et al., 2019). In a three-way classification task, we achieve the macro-F1 score of 0.4027, which is 12% points higher than that of the second-place winner.

Detecting the veracity of rumors spreading out on various social media platforms has been of great importance. Indeed, several studies find that online rumors can affect human behaviors (Pound and Zeckhauser, 1990; Jia et al., 2020). However, detecting the veracity of rumors is not a simple task. Unlike news articles which are considered ex-post, rumors are ex-ante (Vosoughi et al., 2018; Shu et al., 2017). At the time when a rumor originates, the information user is not able to determine its veracity. Instead, the user can make his best guess based on the information set that he has been exposed to. In contrast, we can check the veracity of a news article immediately by comparing it with the event that the article is referring to (Cao et al., 2018).

Whether a rumor is false or not can only be determined afterward when the user can objectively observe the event (Zubiaga et al., 2016). In our research, we use only the textual features of the posts and their corresponding replies, mitigating the concern that our results are driven by external information that was not readily available to the general public at the time of rumor origination. Further, our model shows that textual features embedded in social media posts can reasonably estimate the ex-ante veracity of rumors.

We develop a double-channel model to classify rumors into three categories: true, false, and unverified. First, we assign each thread into a certain category or uncertain category with a BERT-based classifier (Devlin et al., 2018). Posts assigned to the certain category (hereafter certain posts) convey messages in a more confident tone. Authors who make informed guesses are more likely to be confident in their postings (DiFonzo, 2010). As lie detection algorithms systematically identify the author's intention to deceive other people (Mansbach et al., 2021; Barsever et al., 2020), we apply the BERT-based lie detection classifier to the certain posts. Certain posts that are classified as non-lie are true rumors and vice versa.

On the other hand, posts assigned to the uncertain category (hereafter uncertain posts) exhibit less confident tones. Similarly, we presume that authors without access to proprietary information sources are likely to post arbitrary and unsure threads. Therefore, instead of applying lie detection algorithms, we turn to the linguistic features

of primary replies. Users exhibit their opinions regarding the threads via replies, and many prior studies find the remarkable accuracy of the wisdom of the crowds in social media platforms (Brown and Reade, 2019; Navajas et al., 2018). We measure whether the primary replies agree with the main thread. Specifically, we apply the BERT-based agreement-classifier, which assigns each thread-reply pair into the agreement or disagreement category to all the thread-reply pairs. Uncertain posts with many disagreeing primary replies are assigned to the false rumor category and vice versa.

This double-channel approach yields the macro-F1 score of 0.4027, which is approximately 12% points higher than that of the second-place winner. Our research contributes to the existing line of literature for at least two reasons. First, we are the first to employ a double-channel model to detect the veracity of rumors. Even though some use a variety of variables, prior studies fail to obtain state-of-the-art results primarily because they apply the same logic to every post. We propose that lie detection algorithms are relatively more appropriate for classifying certain posts and that an agreement-based classifier is more accurate when classifying uncertain posts. After employing the BERT-based certainty classifier to divide the samples into two subgroups, we find a significant increase in our classification accuracy.

Second, we also use minimal information to obtain our results. Our F1 score falls behind the winner of SemEval 2019 Task 7, primarily due to the scope of the information that we use. The winner exploits a variety of peripheral information such as the account credibility or the number of followers (Li et al., 2019a). We utilize only the textual features extracted from the threads themselves and their primary replies without considering any other information. Even without considering the peripheral or user-specific information, we manage to obtain a reasonable classification accuracy.

2 Related Works

2.1 Information Sets

Prior literature mainly relies on two information sets to calculate the ex-ante veracity of rumors. First, several studies use user information such as the number of followers, the number of replies, the existence of hashtags and photos, and the number of previous tweets to determine the veracity of each rumor (Castillo et al., 2011; Vosoughi, 2015; Liu and Wu, 2018; Li et al., 2019a). This line of research assumes that the users who care about their accounts' reputations are likely to post true rumors. However, it is difficult to measure the account's credibility when the rumor originates since the account information is time-variant. Even though a specific account currently has many followers, we cannot guarantee that the account used to have the same number of followers when the rumor originated. Nonetheless, due to the limitation in data, many studies do not consider the time-varying properties of account information.

Second, several studies apply linguistic features to detect false rumors. Some studies measure the subjectivity of the posts using some attribute-based textual elements such as subjective verbs and imperative tenses (Li et al., 2019a; Ma et al., 2017; Liu et al., 2015). Vosoughi (2015) analyzes the sentiment of tweets under various circumstances and classify the tweets using the contextual information. Barsever et al. (2020) develop a better-performing lie detector with BERT, indicating that unsupervised learning can outperform traditional rule-based lie detection algorithms. However, the linguistic feature-based approach has limitations in that most of the rumors are arbitrary in nature, and lie detection, which is based on the author's intention, may not function well in the domain that contains many random posts.

Other research focuses on the network model to capture information propagation (Gupta et al., 2012; Rosenfeld et al., 2020). Also, Liu and Wu (2018) develop a model that examines the early detection of rumors with RNN classification. However, in our study, we do not consider the spread and propagation of rumors. Also, several works aim to determine whether a given online post is a rumor or not (Kochkina et al., 2018), but such a task exceeds the scope of our research.

2.2 Classification Algorithm

While several studies deal with improving the input dataset, others focus on improving the classification algorithm. The most common classification is based on the Support Vector Machine (SVM) (Enayet and El-Beltagy, 2017; Wu et al., 2015). These studies commonly collect a number of variables regarding each thread and perform the classification. Similarly, several studies employ neural networks (NN) to conduct the classification (Ma et al., 2017; Wang et al., 2018).

However, in our task, we are required to classify not only true and false rumors, but also unverified rumors. Unverified rumors, by definition, are the rumors that are not verifiable ex-post. But, in our program, we set the observations with zero confidence scores to be unverified. Therefore, classification methods such as SVM and NN may not be very accurate in our setting.

Recent works turn to unsupervised learning of rumors. Instead of inputting a number of user-specific variables, Rao et al. (2021) develop STANKER, a fine-tuned BERT model which incorporates both the textual features of posts and their comments. This model inputs comments as one of the crucial auxiliary factors, measuring the co-attention between the posts and comments. Our model differs from STANKER for at least two reasons. First, unlike STANKER which uses single-channel approach, we design a double-channle approach. This approach allows us to apply a more appropriate classifier to each thread. Second, STANKER is trained with more than 5,000 labeled observations. These observations do not include the "unverified" category as well. However, since our train set contains only 365 observations with three different labels, we utilize external open-source datasets from similar (yet slightly different) domains to further train each phase of our model. Therefore, we aim at improving the performance of the model with the minimal information and observations by fine-tuning the model to mitigate the domain-shift problem.

3 Model Design

3.1 Overall Structure

Our model is the first to introduce a double-channel approach in rumor veracity detection. We first divide the sample into two subsamples depending on the certainty score of each thread. Here, a certainty score examines whether the author is writing the post with a strong belief or not (Farkas et al., 2010). Our BERT-based uncertainty-classifier assigns each thread into one of the two categories: certain and uncertain. We assume that certain posts are based on educated belief, insider information, or other reliable sources. Note that when the author has baseline information, it is the author's choice to decide whether or not to disclose the true information to the public. We name this step Phase 1.

Then, we turn to lie detection algorithms for certain posts. Textual lie detection focuses on lexical cues that are prevalent in intentional lies (Masip et al., 2012). Lie detection algorithms examine the author's intention – they identify whether the writer is intentionally distorting the actual information. Therefore, certain posts provide us with an ideal setting to employ the algorithms. If the authors decide to distort their information, a lie detector is expected to identify such intention. We use a BERT-based lie classifier to assign the threads into a true or false category. We call this step Phase 2-1.

On the other hand, for uncertain posts, we cannot rely on linguistic lie detection. Uncertain posts are written by people who do not have any specific reference when spreading the rumors via social media platforms. In other words, they make an uninformed guess or even write some random facts in their accounts. Since the writers do not intend to deceive other people, the lie detection algorithm may not function properly. Therefore, we should take a different approach to determine the veracity of such rumors. Here, we focus on the agreement score of each reply. Users actively respond to the rumors in social media, and the wisdom of the crowd is known to generate remarkably accurate information (Brown and Reade, 2019; Navajas et al., 2018). In our study, we calculate the degree of agreement of each primary reply to the thread. Then, using the agreement score of the replies, we estimate the veracity of the thread. We call this step Phase 2-2. Note that applying the lie detection algorithm to all posts can harm the model's performance since the algorithm mainly captures the writer's intention to deceive people.

For the visual representation of our pipeline, refer to Figure 1. We use Tesla V100 SXM2 32GB GPU to train our model. We also present the list of open-source data sets that we use to train each phase of the model in Table 1.

3.2 Phase 1: Detecting Linguistic Certainty

We develop a BERT-based certainty classifier. Our classifier is a binary classifier based on a BERT sentence classifier. Our goal is to assign each sentence (Twitter or Reddit thread) into one of the two categories: certain or uncertain. We first train our model with the labeled dataset provided in CoNLL-2010 Shared Task (Farkas et al., 2010). The dataset contains binary labels (certain or uncertain) and 7,363 observations. We use a batch size of 32 and

Source Data	Model Trained	Format	Number of Train Data	Labels
SemEval2019 Task7	Main pipeline	Threads, replies	365	True False Unverified
CoNLL2010 Shared Task	Uncertainty classifier (Phase 1)	Threads	7,363	Certain Uncertain
Ott et al. (2011) Ott et al. (2013)	Lie detector (Phase 2-1)	Threads	1,600	True False
Andreas et al. (2012)	Agreement detector (Phase 2-2)	Thread-reply pair	1,163	Agree Disagree None

Table 1: This table reports the list of open-source data sets that we use to train each phase of our model. Since our model comprises of several distinct tasks, we try to find domains and tasks that are similar to our main goals.

a learning rate of 5e-5. We train the model for five epochs and use Adam optimizer.

We apply the trained BERT classifier to our train set. This process yields 365 distinct thread-label pairs. However, the domain of the dataset that we use to train the model slightly differs from the domain of the dataset that we have. To tackle this domain-shift issue, we sample 21 observations from each category (certain and uncertain) and re-train the model for five epochs. We select the same number of observations from the two categories to mitigate the concern arising from severely imbalanced classifications. We use a batch size of 32 and a learning rate of 5e-5. This procedure assuages the potential bias due to domain-shifting.

We set a label smoothing rate of 0.2 for both training steps. Label smoothing resolves the classification imbalance due to the differences in the two domains and the potential overfitting due to the limited number of our training samples (Szegedy et al., 2016). We apply Phase 1 to all test samples and obtain 81 distinct thread-label pairs. 17 of them are classified as certain posts, and the remaining 64 observations are classified as uncertain posts.

3.3 Phase 2-1: Fake Rumor Identification with Lie Detection Algorithm

We apply Phase 2-1 to certain posts from Phase 1. We develop a BERT-based binary sentence classifier to detect lies from lexical cues. Similarly, we take a two-step approach to train the model. First, we use the open-source dataset to train a model that detects scams and lies in social media (Ott et al., 2011; Ott et al., 2013). This dataset contains 1,600 pre-labeled texts. We train the model for five epochs with a batch size of 32, a learning rate of 5e-5, and a label smoothing rate of 0.3. We also use Adam optimizer.

Then, we fine-tune the model with the train dataset of SemEval 2019 Task 7. According to the definition, unverified samples are those with zero confidence scores. Therefore, when fine-tuning our model, unverified observations are of no use. We exclude the unverified samples and use only 221 observations with true or false labels. We train the model for one epoch with a batch size of 32 and a learning rate of 5e-5. Unlike certainty classification of Phase 1, the domains and objectives of the external dataset that we use are similar to our primary goal – determining the veracity of a given statement. However, in Phase 1, the surrogate dataset aims at discerning non-factual and factual information. That is, the objectives of the two tasks are similar but not the same. Therefore, we train the model for five epochs in Phase 1. In Phase 2-1, since the two tasks deal with the same agenda, it suffices to fine-tune the model for one epoch.

When applied to the test set, our lie detector yields 81 distinct thread-label pairs. The label includes true and false indicators based on the softmax values. That is, when the softmax value of true is larger than the softmax of false the program returns true and vice versa. Following the definition of the unverified rumors, we classify the samples with self-entropy score of 1 into unverified category. Otherwise, we use the labels obtained from our lie detector.

We use the following formula to obtain the self-

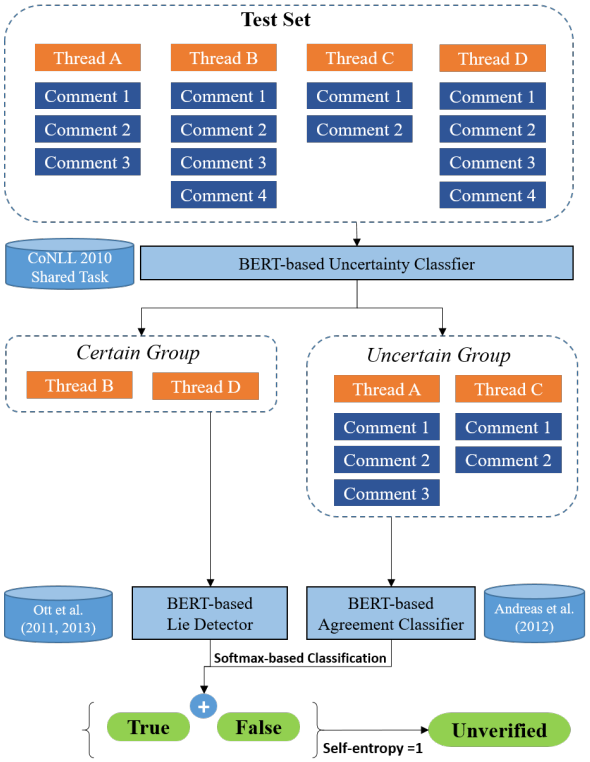

Figure 1: This figure illustrates the model pipeline. Uncertainty classifier (Phase 1) divides the sample into two subgroups, and lie detector (Phase 2-1) and agreement classifier (Phase 2-2) further classifies each thread into true or false category. We assign the observations with self-entropy of 1 to unverified category.

entropy of each observation.

$$H(x) = -\frac{1}{\log 2}\sum_{n=0}^{1} l_n(x)\log l_n(x)$$

, where x denotes each observation and $l_n(x)$ denotes the probability that x belongs to each category ($n = 0, 1$).

3.4 Phase 2-2: Fake Rumor Identification with Reply Agreement Score

We apply Phase 2-2 to uncertain posts from Phase 1. Here, we develop a BERT-based triple sentence classifier that assigns each sentence pair into one of the three categories: agreement, disagreement, and none. Here, the input is a sentence pair composed of one thread and its corresponding primary reply. We exclude non-primary (secondary or tertiary) replies since it is unclear whether such non-primary replies are agreeing (or disagreeing) to the thread itself or the primary reply. Therefore, the classifier measures whether the primary reply is in accordance with the thread or not. We also take a two-step approach to train the model.

First, we train the BERT-based triple classifier with an open-source dataset (Andreas et al., 2012). The dataset contains 1,163 sentence pairs with agreement labels. Specifically, it includes 609 agreement pairs and 554 disagreement pairs. We train the model for five epochs with a batch size of 32, a learning rate of 5e-5, and a label smoothing rate of 0.3. We also use Adam optimizer.

Then, we fine-tune the model with the train set of SemEval 2019 Task 7. We filter out primary responses from the dataset and create thread-reply pairs. We label the pairs with the labels pre-assigned to each thread. This process yields 2,372 distinct thread-reply pairs. Then we train the model for one epoch with batch size 32 and learning rate 5e-5. The task of Andreas et al. (2012) aims at determining whether each reply is in accordance with the thread or not, which is identical to our objective. Therefore, we fine-tune the model for one epoch.

Applying the classifier to uncertain posts yields the softmax values for (agreement, disagreement, none). We discard the softmax value of none and sum the softmax values of agreement and disagreement for each thread. Then, we normalize the values so that they sum up to be one. As in Phase 2-1, the program returns true when the softmax value of the agreement is larger than that of disagreement and vice versa.

For a formal representation, let X_i denote the thread and y_m^i denote the mth primary reply to X_i. Suppose that we have k threads and n_i (i is an integer between 1 and k) is the number of primary comments corresponding to X_i. We form up the pairs $(X_1, y_1^1), \cdots, (X_1, y_{n_1}^2), \cdots, (X_k, y_1^k), \cdots, (X_k, y_{n_k}^k)$. BERT model returns a softmax vector of each pair (a_l, b_l, c_l), where (a, b, c) denotes the softmax vector of (agreement, disagreement, none). We obtain $\sum_{i=1}^{k} n_i$ softmax vectors. Then, for X_i, we sum up the softmax values to obtain the normalized softmax vector.

$$\left(\frac{\sum_{k=1}^{n_i} a_k}{\sum_{k=1}^{n_i} a_k + \sum_{k=1}^{n_i} b_k}, \frac{\sum_{k=1}^{n_i} b_k}{\sum_{k=1}^{n_i} a_k + \sum_{k=1}^{n_i} b_k}\right)$$

If the first softmax is larger than the second, we classify X_i to be true. If the second softmax is larger than the first, we classify X_i to be false.

Also, we assign the observations with the self-entropy value of 1 to the unverified category. We calculate the self-entropy using the same formula with Phase 2-1.

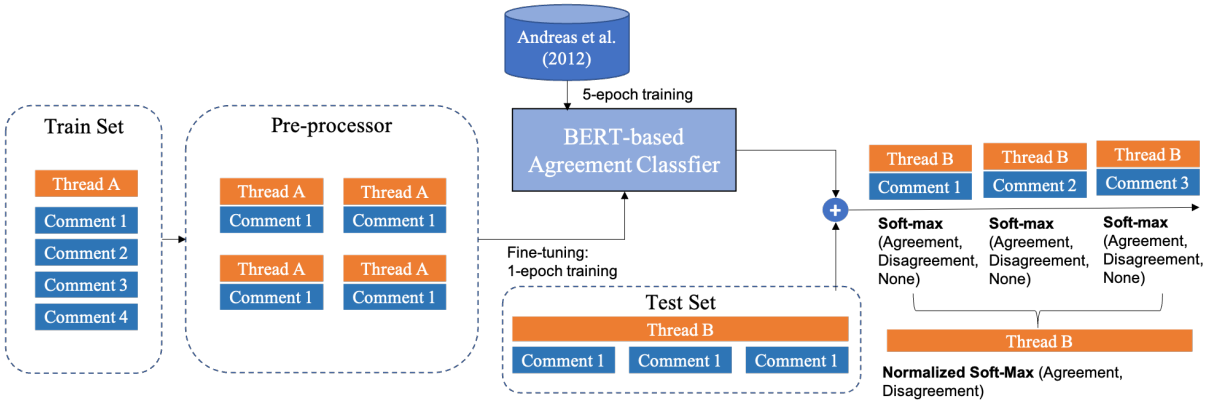

Figure 2: This figure illustrates the pipeline of Phase 2-2. We pre-train the BERT model with the dataset provided by Andreas et al. (2012) and fine-tune the model with pre-processed train set of SemEval 2019 Task 7. Then we apply the BERT-based agreement detector to thread-reply pair of the test set and obtain soft-max value vectors. We discard the soft-max values of *none* since *none* does not provide additional information about the veracity of the rumors.

We discard none because replies that do not fall under either agreement or disagreement category do not have informational value. By allowing the none category and discarding the none category samples, we aim to deliberately examine the replies' intent (Li et al., 2019a). Refer to Figure 2 for the graphical illustration of Phase 2-2.

In summary, our double-channel model functions as follows:

- Input the test set of SemEval 2019 Task 7 which comprises of Twitter and Reddit posts.
- Phase 1 of the model classifies each post into one of the following categories: uncertain and certain.
  - For the posts categorized in the certain category, apply Phase 2-1 (lie detection). The observations with a "true" label are classified as true rumors. On the other hand, the remaining observations with a "false" label are classified as false rumors.
  - For the posts categorized in the uncertain category, sort out the primary replies and form thread-reply pairs. Phase 2-2 (agreement detection) classifies each pair into one of the following categories: agree, disagree, and none.
  - Discard the *none* samples and calculate the normalized agreement score of each thread based on the softmax values of agree and disagree of each pair.
- Calculate the self-entropy of each observation and assign "unverified" to the samples with the self-entropy value of 1.

3.5 Data and Pre-processing

Our primary input data is the open-source data released in SemEval 2019 Task 7. Specifically, we aim to improve the model performance of Task 7B, in which the participants are asked to verify the true or false of each rumor. The dataset contains 365 train set observations. Each observation consists of one thread (Twitter or Reddit) post and its corresponding replies. Replies include the primary replies (replies that respond directly to the main post) and secondary replies (replies that respond to other replies). In our task, we do not use replies other than primary replies.

We first retrieve all main posts (threads) from the dataset. The threads often include hashtags or web addresses starting with html. Several studies including Li et al. (2019a) use this as auxiliary information in their analysis - they include an indicator variable that equals one when the thread has a hashtag or web address inside. However, in our research, we focus only on textual features and do not need such information. Further, given that the threads are relatively short, uninterpretable hashtags or web addresses might distort the results. Therefore, we delete all hashtags and web addresses that start with "html".

Then, we turn to the comments. The dataset contains a structure file in json format for each thread. The structure file explains the format of

	Macro F1 Score	Accuracy
Double-Channel	**0.4027**	**0.4938**
Single-Channel (Lie Detector)	0.3447	0.4444
Single-Channel (Agreement Detector)	0.3668	0.4444
Baseline (LSTM)	0.3364	-
Baseline (NileTMRG)	0.3089	-
Baseline (Majority class)	0.2241	-
WeST (CLEARumor)	0.2856	-

Table 2: This table demonstrates the relative performances of the models that we develop, the baseline models of SemEval 2019 Task 7, and the second-place winner of the task (WeST). Single-channel models include the model that applies lie detector to all observations and the model that applies agreement detector to all observations.

each thread such as how many comments are there, the time when each comment is posted, the ID of the author and the ID of the comment. From the json file, we identify the primary comments and pair them with their corresponding thread. We also cleanse the texts by removing all the hashtags and web addresses.

4 Results

We present our results in Table 2. We report two evaluation metrics, macro-F1 and accuracy. Macro-F1 is the harmonic average of the precision and recall ratios, while accuracy is the ratio of correct classifications to the total number of observations.

Our double-channel model achieves a macro-F1 score of 0.4027 and an accuracy of 0.4938.[1] This model outperforms all the baseline models proposed in SemEval 2019 Task 7 and the model developed by the second-place winner. Note that our program only refers to textual information of the main threads and their primary replies. We intentionally do not include user-specific peripheral information to demonstrate that the double-channel approach can significantly improve the classification outcomes. In support of our claim, we also report the results when we apply Phase 2-1 and Phase 2-2 classifiers to all observations without the primary classification of Phase 1. The results yield the macro-F1 scores of 0.3447 and 0.3668, respectively. As clearly indicated, dividing the total sample into two subgroups significantly improves the classification performance. This improvement is primarily because each classifier is applied to the observations that the classifier is intended to function well.

Our model outperforms the second-best program (WeST) by approximately 12% points in terms of macro-F1. There is a huge gap in performance between the best performer (macro-F1 = 0.57) and the second-best performer (macro-F1 = 0.28). This is primarily due to the following two reasons:

1. The number of train and test observations is relatively small. The task only provides 365 train observations and 81 test observations. Further, the task requires to classify "*unverified*" rumors as well. Therefore, it is extremely difficult to find an external source of labeled information in parallel with this dataset.

2. While the best performer (eventAI) incorporates a number of different variables, other studies tend to focus on one dimension or feature.

Our model, even though it focuses only on textual dimension of the threads and their comments, achieve a significantly higher macro-F1 than the second-best model. With the double-channel classification system that we develop, we manage to accurately classify false rumors at their early stage, without considering the peripheral information sets.

Our model falls behind the winner of SemEval 2019 Task 7, primarily because we use limited scope of information. We intentionally discard all other information but textual information of the threads and their primary replies. In contrast, the winner exploits a wide variety of information such as account credibility and the existence of hashtags. Our approach differs from this approach in that we aim to suggest preliminary evidence that double-channel classification produces more accurate results than single-channel classifications in rumor veracity detection tasks.

[1] The model correctly classifies 19 true rumors out of 31, 20 false rumors out of 40, and 1 unverified rumor out of 10.

5 Conclusion

Perfectly determining the veracity of rumors at the time of their origination is impossible. Nonetheless, an increasing number of rumors are spreading out via social media, and people are affected by those rumors. Therefore, sorting out the "likely-fraudulent" rumors is of great importance to information users.

Our model takes minimal textual information and achieves a reasonable prediction accuracy in the SemEval 2019 Task 7 dataset. This dataset contains only 365 train samples and 81 test samples and requires three-way classification. We achieve the macro-F1 score of 0.4027 in this task, which is approximately 12% points higher than that of the second-place winner.

Instead of integrating a wide variety of user-specific information, our model shows that textual features have sufficient predictive power in determining the veracity of rumors. More importantly, we demonstrate that applying a uniform classifier to all Twitter and Reddit posts can harm the model's performance. Focusing on the idea that lie detection is intended to sort out the counterfactual statements based on the writers' intention, we are the first to apply a double-channel approach in rumor veracity detection. We divide the sample into two subgroups depending on the textual certainty and apply two different classifiers to each subgroup. Also, by using only textual features of a post and its primary replies, this study responds to Li et al. (2019b)'s call for research that enables the early detection of rumors and exploits target users' response in veracity detection.

Our research opens up a broad potential for future works as well. Our study does not include user-specific information to show that we can achieve better performance with minimal textual information. However, in future work, one may use account credibility information as a weight in training the model. As shown in Li et al. (2019a), such information may help boost the accuracy of the classification model.

References

Jacob Andreas, Sara Rosenthal, and Kathleen R McKeown. 2012. Annotating agreement and disagreement in threaded discussion. In *LREC*, pages 818–822. Citeseer.

Dan Barsever, Sameer Singh, and Emre Neftci. 2020. Building a better lie detector with bert: The difference between truth and lies. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Alasdair Brown and J James Reade. 2019. The wisdom of amateur crowds: Evidence from an online community of sports tipsters. *European Journal of Operational Research*, 272(3):1073–1081.

Juan Cao, Junbo Guo, Xirong Li, Zhiwei Jin, Han Guo, and Jintao Li. 2018. Automatic rumor detection on microblogs: A survey. *arXiv preprint arXiv:1807.03505*.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

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

Nicholas DiFonzo. 2010. Ferreting facts or fashioning fallacies? factors in rumor accuracy. *Social and Personality Psychology Compass*, 4(11):1124–1137.

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.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The conll-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the fourteenth conference on computational natural language learning–Shared task*, pages 1–12.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. Semeval-2019 task 7: Rumoureval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854.

Manish Gupta, Peixiang Zhao, and Jiawei Han. 2012. Evaluating event credibility on twitter. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 153–164. SIAM.

Weishi Jia, Giulia Redigolo, Susan Shu, and Jingran Zhao. 2020. Can social media distort price discovery? evidence from merger rumors. *Journal of Accounting and Economics*, 70(1):101334.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*.

Quanzhi Li, Qiong Zhang, Luo Si, and Yingchi Liu. 2019a. Rumor detection on social media: datasets, methods and opportunities. *arXiv preprint arXiv:1911.07199*.

Quanzhi Li, Qiong Zhang, Luo Si, and Yingchi Liu. 2019b. Rumor detection on social media: datasets, methods and opportunities. *arXiv preprint arXiv:1911.07199*.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1867–1870.

Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-second AAAI conference on artificial intelligence*.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Noa Mansbach, Evgeny Hershkovitch Neiterman, and Amos Azaria. 2021. An agent for competing with humans in a deceptive game based on vocal cues. *Proc. Interspeech 2021*, pages 4134–4138.

Jaume Masip, Maria Bethencourt, Guadalupe Lucas, MIRIAM SÁNCHEZ-SAN SEGUNDO, and Carmen Herrero. 2012. Deception detection from written accounts. *Scandinavian Journal of Psychology*, 53(2):103–111.

Joaquin Navajas, Tamara Niella, Gerry Garbulsky, Bahador Bahrami, and Mariano Sigman. 2018. Aggregated knowledge from a small number of debates outperforms the wisdom of large crowds. *Nature Human Behaviour*, 2(2):126–132.

Myle Ott, Claire Cardie, and Jeffrey T Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 497–501.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*.

John Pound and Richard Zeckhauser. 1990. Clearly heard on the street: The effect of takeover rumors on stock prices. *Journal of Business*, pages 291–308.

Dongning Rao, Xin Miao, Zhihua Jiang, and Ran Li. 2021. Stanker: Stacking network based on level-grained attention-masked bert for rumor detection on social media. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3347–3363.

Nir Rosenfeld, Aron Szanto, and David C Parkes. 2020. A kernel of truth: Determining rumor veracity on twitter by diffusion pattern alone. In *Proceedings of The Web Conference 2020*, pages 1018–1028.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Soroush Vosoughi. 2015. *Automatic detection and verification of rumors on Twitter*. Ph.D. thesis, Massachusetts Institute of Technology.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857.

Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In *2015 IEEE 31st international conference on data engineering*, pages 651–662. IEEE.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.