Leveraging Social Interactions to Detect Misinformation on Social Media

Tommaso Fornaciari[1], Luca Luceri[2], Emilio Ferrara[3], Dirk Hovy[4]
[1]Italian National Police
[2, 3]University of Southern California
[4]Bocconi University
[1]tommaso.fornaciari@poliziadistato.it, [2]lluceri@isi.edu,
[3]emiliofe@usc.edu, [4]dirk.hovy@unibocconi.it

Abstract

Detecting misinformation threads is crucial to guarantee a healthy environment on social media. We address the problem using the data set created during the COVID-19 pandemic. It contains cascades of tweets discussing information weakly labeled as *reliable* or *unreliable*, based on a previous evaluation of the information source. The models identifying unreliable threads usually rely on textual features. But reliability is not just *what* is said, but by whom and to whom. We additionally leverage on network information. Following the homophily principle, we hypothesize that users who interact are generally interested in similar topics and spreading similar kinds of news, which in turn is generally reliable or not. We test several methods to learn representations of the social interactions within the cascades, combining them with deep neural language models in a Multi-Input (MI) framework. Keeping track of the sequence of the interactions during the time, we improve over previous state-of-the-art models.

Introduction

Social media networks allow the wide and fast diffusion of pieces of information, news, and opinions among interacting users. However, during the last decade, the veracity and accuracy of the shared content have been largely undermined by various factors, including fake accounts and orchestrated disinformation campaigns. Fact-checking the reliability of the shared messages represents nowadays a fundamental need to preserve the integrity of online discussions and healthy fruition of social media services. Automatically detecting misinformation spreading on social media is, however, a challenging task, as proved by the research community (Sharma et al. 2019). Existing solutions show promising results in the classification of *reliable* and *unreliable* content leveraging the text of the shared messages. Here we identify the threads on Twitter according to the notion of *cascade*, as defined by Yang and Leskovec (2010): a sequence of reciprocally engaged tweets, ordered by their time-stamp, starting from a source post at the origin of the sequence. We denote the cascades' reliability and unreliability according to the data set of Sharma, Ferrara, and Liu (2022), where the decision is mainly taken relying on an *a priori* reliability evaluation of the source that issued the first tweet of the cascade.

Contributions In this paper we combine pretrained-Language Models and network-based methods, in previous literature applied to other tasks, to identify unreliable tweet cascades. We reach new State-Of-The-Art performance levels for this task. We show how unreliable news are 1) generally associated with different communities, which can be identified and leveraged for inference, and 2) blended with reliable news in content and style, which might not necessarily carry a strong signal.

Related work

In the last ten years, several methods have been applied to the identification of unreliable cascades. Kumar and Geethakumari (2014) followed a cognitive psychology approach. Zhang et al. (2016) work on time constraints to identify them. Yu et al. (2017) rely on textual data and propose the use of a convolutional neural network, similarly to Zhou et al. (2020) who address stylometry and linguistic feature engineering to detect fake news. Also Monti et al. (2019) rely on convolutional neural networks, but they are interested in propagation patterns, that is the geometry of the social networks that share news. A similar, hierarchical approach is followed by Shu et al. (2020). Deep Learning methods, such as LSTM, are applied to the texts by Ducci, Kraus, and Feuerriegel (2020), Pierri, Piccardi, and Ceri (2020). This approach is similar to that applied by Sharma, Ferrara, and Liu (2022), who used the CSI model of Ruchansky, Seo, and Liu (2017), which employs a recurrent neural network to represent texts and user behaviors. To capture social interactions, we use mentions2vec - M2V, a method proposed by Fornaciari and Hovy (2019) and applied on a geolocation task. Guess, Nagler, and Tucker (2019) study the 2016 U.S. presidential campaign, identifying specific communities sharing fake news: their findings fit with our results, as discussed in Section User clustering. Ma, Gao, and Wong (2017) and Rosenfeld, Szanto, and Parkes (2020) reach similar conclusions applying Tree Kernel to the propagation structure of news; their studies are complementary to this paper, as they are interested in the cascade structure, while we focus on the involved accounts. Similarly to our work, Nguyen et al. (2020) create user vector representations; however they rely on TF-IDF (Term Frequency - inverse Document Frequency) measures from the users' profile: in contrast we leverage on their social interaction within the news cascades. Lu and

Li (2020) propose a complex multi-input neural approach, where the users are represented by their metadata, and their relationships are modeled considering their presence in the same cascade. We address their interactions as they develop directly in their posts (potentially catching signal from users not active in a cascade, but simply mentioned by someone).

Tweets kind	Amount
Retweeted tweet without comment	328 521
Quoted tweet	32 218
Reply	10 161
Original	5 328
Total	376 228

Table 1: Corpus statistics.

Data

The data set of Sharma, Ferrara, and Liu (2022), collected during the COVID-19 pandemic, from December 9th, 2020 to February 24th, 2021, contains 14 644 cascades (10 377 reliable, 4 267 unreliable), already divided in training, development and test set. The cascades contain 376 228 tweets, issued by 168 227 users. The texts are in English and already pre-processed. Each cascade contains an average of 25.68 tweets; more details in Table 1.

Methods

We implement five different models to detect misinformation tweet cascades. The first is the baseline, to which we compare the four other models. All models perform the same classification task. The baseline model is a text-only Single-Input BERT-based (Devlin et al. 2018) model. It uses the contextual word embeddings from BERT, without fine-tuning the whole BERT. In particular, we use the mean of the word vectors of the concatenated tweets from the whole cascade. However, between BERT's output and the standard, fully-connected classification layer, we insert a further Transformer mechanism (Vaswani et al. 2017). This approach has proven more effective than using a fully-connected output layer alone, in several NLP tasks (Fornaciari et al. 2021).

Similarly to Sharma, Ferrara, and Liu (2022), who use the same kind of inputs, we explore four different types of Multi-Input models, fed with different combinations of textual and network-interaction information. The textual data are represented via the BERT-based language model, like in the baseline model. The network interactions are encoded via three different methods, as follows.

Multi-Input: network-sparse-vectors The simplest way to represent a cascade in a social network as a vector is to encode all users' presence or absence in each tweet cascade. To keep the vectors within a manageable size and reduce the noise from uninformative data (e.g., cascades with few or infrequent users), we only considered users that performed at least 15 actions (i.e., tweets, retweets, replies, or quotes) in one or more cascades in the dataset. We chose the threshold of 15 based on computational affordability (see last paragraph in this Section). This method produces sparse vectors of size 1326 for each cascade. The dimensions correspond to the 1326 selected users, with the values 1 if the user is present in the cascade, and 0 otherwise. In this model, both the textual (BERT) and network (sparse) representations are separately fed into two Transformers (Vaswani et al. 2017), whose outputs are concatenated and passed to the final classification layer.

Multi-Input: network-embeddings In the second model, the sparse vectors conveying the network interactions view are not passed directly to an attention mechanism but are fed into a fully connected layer, which squeezes them into dense vectors of size 128. These smaller, dense vectors can be considered learned (i.e., trainable) network embeddings. Then, similarly to the previous models, network and text (BERT) embeddings are passed to two parallel attention mechanisms connected to the classification layer.

Multi-Input: mentions2vec - M2V network-embeddings In the third model, we again use the textual BERT representations as in the previous models. For the network representation, we use mentions2vec, a methods based on Doc2Vec (Le and Mikolov 2014) proposed by Fornaciari and Hovy (2019) (there to improve model performance in a geolocation task). M2V filters the texts to preserve only the users' mentions (i.e., user names starting with "@"). This procedure results in "texts" containing only sequences of users' mentions. In this way, the texts represent explicit social interactions on social media. These sequences are then encoded as dense vectors using Doc2Vec. Doc2Vec allows for the assignment of document labels, typically the document ID. Here, we substituted this label with the cascade ID. This procedure has a critical advantage over the other, traditional methods of network representation. Those rely on square (adjacency) matrices that grow quadratically with the network size, a constraint that quickly becomes computationally unsustainable. Therefore, other methods need to keep rigid control of the network size and typically revert to some form of sampling when the network size becomes too large. M2V, in contrast, produces fixed-length vectors of a chosen size, independent of network size. The number of users does not affect the size of the representations, and the number of user mentions acts as a "vocabulary" in Doc2Vec. This way the growth of the network representation is linear with the number of texts, rather than quadratic with the number of users. We feed the M2V network embeddings into the same architecture used for the previous experimental models.

Multi-Input: retrofitted-BERT and network-embeddings In the last model, we use the same network-embedding representations as before. However, we evaluate the possibility of injecting cascade information into the BERT embeddings. We do this via the cascade classes in the training set, and use them in the retrofitting method proposed by Faruqui et al. (2015). It forces the vectors of instances belonging to the same equivalence class (here, the cascade class) to be more similar to each other, thereby increasing the distance between instances of different classes. This kind of transformation can be reproduced on unseen data, even if the class is unknown,

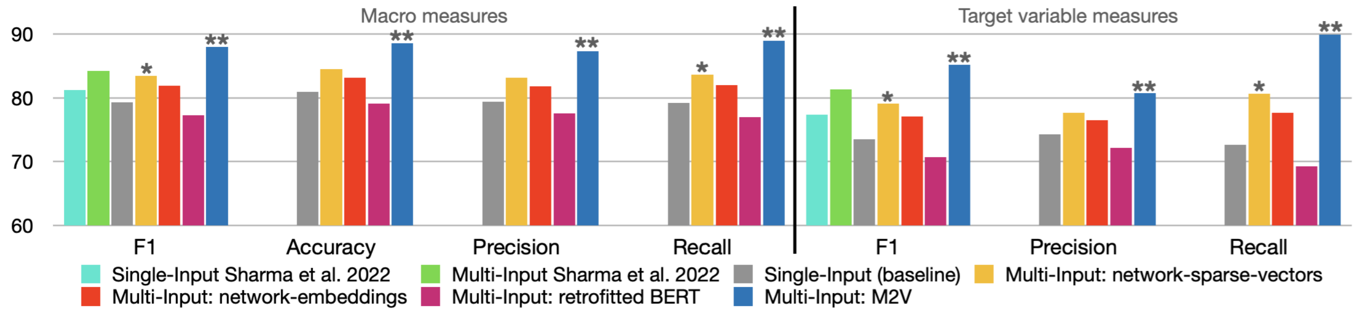

Figure 1: Overall and target class (i.e., unreliable cascades) performance. Significance: $^{**}: p \leq 0.01$; $^{*}: p \leq 0.05$.

using a translation matrix that approximates the original operation of matrix transformation (Faruqui et al. 2015; Hovy and Fornaciari 2018). In our case, we retrofit the texts' representations of the training data according to their relative cascade label. I.e., we start with BERT mean word embeddings, and increase the similarity of all vectors that represent reliable cases, and the similarity of all vectors labeled unreliable. Then we learn a translation matrix from the original BERT embeddings to the retrofitted ones, and apply the translation to the development and test data to create a retrofitted version of them. This second step does no longer require access to labels: the translation matrix has learned how to transform textual embeddings to reflect cascade classes.

Retrofitting the data is a form of pre-processing, as it precedes training and is not affected by the model training. We use the same neural architecture of the previous model, but fed with retrofitted rather than standard textual embeddings. Strictly speaking, this procedure does not leverage network information directly, as it relies on cascade labels. However, this approach lets us verify whether we can improve model performance by leveraging the association between texts based on their labels.

Computational load, parameters and hyper-parameters. In our experiments, we used a GPU NVIDIA GeForce GTX 1080 Ti. The BERT models were not fine-tuned. The number of trainable parameters ranged from 8.5M for the Single-Input models to 20M for the mentions2vec-based models. To reduce the random initializations' impact, we carried out five experiments for each experimental condition. The creation of the BERT text representation took 40 minutes, and the set of the following experiments was approximately one hour. The training was stopped with an early stopping algorithm, relying on the development set's F-measure. The tables in Appendix show the mean epochs for each experimental condition. We used Transformers with one layer and one head, dropout probability for the classification layer at 0.1. These hyper-parameters were found through empirical search.

Results

Figure 1 show the results. The left side shows the macro performance, that is the overall performance averaged over the two classes. The right side focuses on the target class, that is the performance on predicting unreliable cascades. In both cases, we compare our results to the performance of the best previously-reported models (Sharma, Ferrara, and Liu 2022). Following common good practice in NLP, we use bootstrap sampling (Efron and Tibshirani 1994; Berg-Kirkpatrick, Burkett, and Klein 2012) to compute the performance significance between the Multi-Input models and the Single-Input baseline. We repeat 1000 tests per model, with a sampling size of 30% (Søgaard et al. 2014; Fornaciari et al. 2022). The models of Sharma, Ferrara, and Liu (2022) are challenging to beat. Their Single-Input model handily beats our corresponding baseline model. Their Multi-Input model tends to be better than most formulations we explore. However, our Multi-Input model using M2V manages to significantly improve over the baseline with $p \leq 0.01$, and it improves by more than 3.5 points F1 over the best model of Sharma, Ferrara, and Liu (2022) in both settings (the macro value and the target class only, see Appendix). By comparing our Multi-Input models against the Single-Input ones, which all share the same textual representation, we can measure the specific network representation's contribution to the classification task. The models relying on network-sparse-vectors are significantly better than our Single-Input baseline. The models with network embeddings are still better than the control, but not significantly.

Lastly, the models that use retrofitted BERT embeddings show results that are even worse than the baseline, which did not incorporate network information.

Discussion

The low performance of the model with retrofitted BERT embeddings is an interesting result. Making the cascade representations from the same label class more similar does not improve performance. This outcome suggests that, from the point of view of style and content, reliable and unreliable cascades are quite similar to each other. *Ex post*, it makes sense that topics completely different from each other could still share the same feature of being reliable or unreliable. In contrast, network representations are clearly useful for classification. We assume that different communities are prone to congregate around different topics that tend to be systematically more reliable or unreliable. Feeding sparse cascade vectors directly into the Transformers is a strategy more effective than previously reducing their dimension with a dense representation. Since the dense representation approximates the sparse one, the results are not surprising. However, for

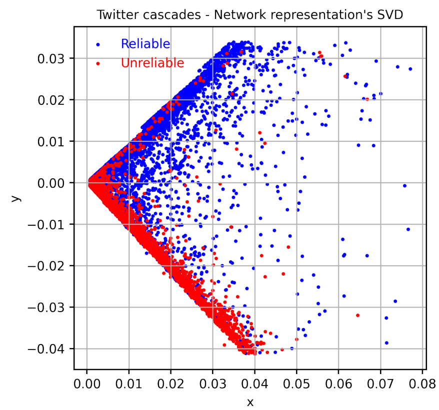

Figure 2: Twitter cascades in the test data represented by their users via SVD. Color shows label class.

wider networks, feeding sparse vectors into Transformers that rely on multiple ‘key’, ‘query’, and ‘value’ square matrices could be computationally unaffordable (Vaswani et al. 2017). Finally, the M2V approach proves particularly effective for the task. The information modeled in this representation is much richer than that simply inferred by counting the users present in the same cascade. M2V considers all the accounts a user addresses in the texts that he/she produces. This set of accounts can also include ‘silent’ users and so can be (much) wider than the group of users who actively participate in the cascade. This means that the social representation is particularly expressive. Also, users can be mentioned several times, which would give their presence (or influence) more weight in the representation.

User/cascade clustering

To test our hypothesis that reliable and unreliable cascades are really fed by different communities, we applied unsupervised methods to cluster the cascades according to the users who wrote texts in the cascades. In particular, we vectorized the cascades via the method used to create the sparse network representation, but without filtering the users according to the frequency threshold of 15, used in that case (Section Methods, paragraph Multi-Input: network-sparse-vectors). Then, we reduced the vectors’ dimensionality with Truncated Singular Value Decomposition - SVD (Sanderson 2010, Chapter 18). The results, without outliers (which would have “zoomed out” the whole image), are shown in Figure 2. Reliable and unreliable cascades are clearly positioned in different regions of the chart, suggesting that they are characterized by the presence of different users. This finding points towards a community-driven aspect of reliability. Following the low results from retrofitted BERT embeddings, we also examined the content overlap between reliable and unreliable cascades. We applied to the cascades PROgressive SImilarity Thresholds - ProSiT, a recent method for topic models (Fornaciari, Hovy, and Bianchi 2022). Given a corpus represented in the same vector space, ProSit finds latent dimensions that are considered as topics. ProSiT does not require the user to provide a wanted number of topics: it takes a similarity threshold

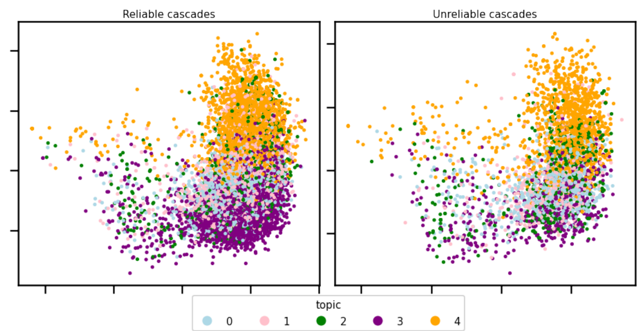

Figure 3: SVD on reliable and unreliable cascades. The point colors represent the most representative topic for the cascade

Descriptors	% cascade rel.	unrel.
0) people, worker, health, vaccinate, healthcare	17.40	21.16
1) variant, south, effective, africa, astrazeneca	17.92	12.59
2) uk, eu, brexit, europe, jab	9.80	12.23
3) administration, trump, dose, state, government	29.35	14.60
4) covid, doctor, trial, die, pharmacist	25.53	39.42

Table 2: Topics from ProSiT and their distribution within reliable and unreliable cascades.

as a hyper parameter, and it presents different hypotheses of topics, among which the most meaningful can be selected by standard metrics or directly by the reader. We identified five main topics, whose descriptors are shown in Table 2, with the rate of reliable and unreliable cascades mainly associated to each of them. We also applied SVD to the cascade textual representations. In this case, for better topic readability, we only consider content words, removing function words. Figure 3 shows reliable and unreliable cascades, where the colors represent the topic having the highest affinity with each of them. While some topics are more frequently associated with reliable cascades (e.g., topic 3) or unreliable cascades (e.g. topic 4), it is also clear that the cascade reliability/unreliability is a transversal feature across every topic.

Conclusion

In this paper, we explored four computational methods to detect unreliable tweet cascades. Our results suggest that these harmful threads can contain various topics; however, they mostly are generated by distinct communities. Therefore it is useful to support the linguistic representations with a network view, which proves effective for this task. Among other methods, we find that mentions2vec (Fornaciari and Hovy 2019) is an efficient way to encode user interactions within the cascades. As recent research demonstrates that some users play a pivotal role in diffusing questionable information (Nogara et al. 2022; Yang et al. 2021; DeVerna et al. 2022), in future work, we will develop solutions to embed in our models the activity of the so-called misinformation “superspreaders”.

Limitations

English is the target language of this study. Reproducibility might be problematic with languages with wider morphology. Also, the presented methods of social network analysis require data from social media that allow mentioning unambiguously other user accounts with some markup sign, "@" in the Twitter case.

Ethical Considerations

We adopted publicly available datasets for training and testing our framework but we did not devote sufficient time and attention to the possible biases that our model might have that could yield practical implications in real-world applications. We do not believe our framework is harmful *per se*. However, as the input documents and their representations might carry biases, unethical content, and/or personal information, issues of fairness, bias, and data privacy might arise. Therefore, we invite further research and responsible use of this framework.

Acknowledgments

Work supported in part by DARPA (contract #HR001121C0169).

References

Berg-Kirkpatrick, T.; Burkett, D.; and Klein, D. 2012. An Empirical Investigation of Statistical Significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 995–1005. Jeju Island, Korea: Association for Computational Linguistics.

DeVerna, M. R.; Aiyappa, R.; Pacheco, D.; Bryden, J.; and Menczer, F. 2022. Identification and characterization of misinformation superspreaders on social media. *arXiv preprint arXiv:2207.09524*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ducci, F.; Kraus, M.; and Feuerriegel, S. 2020. Cascade-LSTM: A Tree-Structured Neural Classifier for Detecting Misinformation Cascades. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '20, 2666–2676. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379984.

Efron, B.; and Tibshirani, R. J. 1994. *An introduction to the bootstrap*. CRC press.

Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E.; and Smith, N. A. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1606–1615.

Fornaciari, T.; Bianchi, F.; Nozza, D.; and Hovy, D. 2021. MilaNLP @ WASSA: Does BERT Feel Sad When You Cry? In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 269–273. Online: Association for Computational Linguistics.

Fornaciari, T.; and Hovy, D. 2019. Dense Node Representation for Geolocation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 224–230. Hong Kong, China: Association for Computational Linguistics.

Fornaciari, T.; Hovy, D.; and Bianchi, F. 2022. ProSiT! Latent Variable Discovery with PROgressive SImilarity Thresholds. arXiv:2210.14763.

Fornaciari, T.; Uma, A.; Poesio, M.; and Hovy, D. 2022. Hard and Soft Evaluation of NLP models with BOOtSTrap SAmpling - BooStSa. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 127–134. Dublin, Ireland: Association for Computational Linguistics.

Guess, A.; Nagler, J.; and Tucker, J. 2019. Less than you think: Prevalence and predictors of fake news dissemination on Facebook. *Science advances*, 5(1): eaau4586.

Hovy, D.; and Fornaciari, T. 2018. Increasing In-Class Similarity by Retrofitting Embeddings with Demographic Information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 671–677. Brussels, Belgium: Association for Computational Linguistics.

Kumar, K.; and Geethakumari, G. 2014. Detecting misinformation in online social networks using cognitive psychology. *Human-centric Computing and Information Sciences*, 4(1): 1–22.

Le, Q.; and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196.

Lu, Y.-J.; and Li, C.-T. 2020. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv preprint arXiv:2004.11648*.

Ma, J.; Gao, W.; and Wong, K.-F. 2017. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 708–717. Vancouver, Canada: Association for Computational Linguistics.

Monti, F.; Frasca, F.; Eynard, D.; Mannion, D.; and Bronstein, M. M. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Nguyen, V.-H.; Sugiyama, K.; Nakov, P.; and Kan, M.-Y. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 1165–1174.

Nogara, G.; Vishnuprasad, P. S.; Cardoso, F.; Ayoub, O.; Giordano, S.; and Luceri, L. 2022. The Disinformation Dozen: An Exploratory Analysis of Covid-19 Disinformation Proliferation on Twitter. In *14th ACM Web Science Conference 2022*, 348–358.

Pierri, F.; Piccardi, C.; and Ceri, S. 2020. A multi-layer approach to disinformation detection in US and Italian news spreading on Twitter. *EPJ Data Science*, 9(1): 35.

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

Ruchansky, N.; Seo, S.; and Liu, Y. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, 797–806. New York, NY, USA: Association for Computing Machinery. ISBN 9781450349185.

Sanderson, M. 2010. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. ISBN-13 978-0-521-86571-5, xxi+ 482 pages. *Natural Language Engineering*, 16(1): 100–103.

Sharma, K.; Ferrara, E.; and Liu, Y. 2022. Construction of Large-Scale Misinformation Labeled Datasets from Social Media Discourse using Label Refinement. In *Proceedings of the ACM Web Conference 2022*, 3755–3764.

Sharma, K.; Qian, F.; Jiang, H.; Ruchansky, N.; Zhang, M.; and Liu, Y. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3): 1–42.

Shu, K.; Mahudeswaran, D.; Wang, S.; and Liu, H. 2020. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the international AAAI conference on web and social media*, volume 14, 626–637.

Søgaard, A.; Johannsen, A.; Plank, B.; Hovy, D.; and Martínez Alonso, H. 2014. What's in a p-value in NLP? In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 1–10. Ann Arbor, Michigan: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yang, J.; and Leskovec, J. 2010. Modeling information diffusion in implicit networks. In *2010 IEEE International Conference on Data Mining*, 599–608. IEEE.

Yang, K.-C.; Pierri, F.; Hui, P.-M.; Axelrod, D.; Torres-Lugo, C.; Bryden, J.; and Menczer, F. 2021. The COVID-19 Infodemic: Twitter versus Facebook. *Big Data & Society*, 8(1): 20539517211013861.

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; Tan, T.; et al. 2017. A Convolutional Approach for Misinformation Identification. In *IJCAI*, 3901–3907.

Zhang, H.; Kuhnle, A.; Zhang, H.; and Thai, M. T. 2016. Detecting misinformation in online social networks before it is too late. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 541–548.

Zhou, X.; Jain, A.; Phoha, V. V.; and Zafarani, R. 2020. Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, 1(2): 1–25.

Appendix

Model	Mean epochs	Macro-F1	Std. Dev.	Accuracy	Precision	Recall
Single-Input Sharma, Ferrara, and Liu (2022): Weak labels		81.20	0.01			
Multi-Input Sharma, Ferrara, and Liu (2022): Social+Detection model		84.20	0.01			
Single-Input (baseline)	14.20	79.30	0.01	80.94	79.45	79.17
Multi-Input: network-sparse-vectors	13.20	83.42 *	0.02	84.53	83.17	83.70 *
Multi-Input: network-embeddings	23.60	81.90	0.01	83.20	81.81	82.01
Multi-Input: retrofitted BERT + network-embeddings	11.40	77.25	0.01	79.14	77.53	77.02
Multi-Input: M2V	10.00	**87.95 ****	0.00	**88.59 ****	**87.34 ****	**88.92 ****

Table 3: Overall performance on detection task. Significance: $^{**}: p \leq 0.01$; $^{*}: p \leq 0.05$. **Bold**: best column result.

Model	Mean epochs	F1	Std. Dev.	Precision	Recall
Single-Input Sharma, Ferrara, and Liu (2022): Weak labels		77.40	0.02		
Multi-Input Sharma, Ferrara, and Liu (2022): Social+Detection model		81.30	0.01		
Single-Input (baseline)	14.20	73.48	0.02	74.29	72.69
Multi-Input: network-sparse-vectors	13.20	79.11 *	0.03	77.64	80.65 *
Multi-Input: network-embeddings	23.60	77.05	0.02	76.48	77.63
Multi-Input: retrofitted BERT + network-embeddings	11.40	70.69	0.02	72.20	69.25
Multi-Input: M2V	10.00	**85.16 ****	0.00	**80.73 ***	**90.11 ****

Table 4: Performance on detecting target class *only* (i.e., unreliable cascades). Significance: $^{**}: p \leq 0.01$; $^{*}: p \leq 0.05$. **Bold**: best column result.