Interactively Learning Social Media Representations Improves News Source Factuality Detection

Nikhil Mehta
Department of Computer Science
Purdue University
West Lafayette, IN 47907
mehta52@purdue.edu

Dan Goldwasser
Department of Computer Science
Purdue University
West Lafayette, IN 47907
dgoldwas@purdue.edu

Abstract

The rise of social media has enabled the widespread propagation of fake news, text that is published with an intent to spread misinformation and sway beliefs. Rapidly detecting fake news, especially as new events arise, is important to prevent misinformation.

While prior works have tackled this problem using supervised learning systems, automatedly modeling the complexities of the social media landscape that enables the spread of fake news is challenging. On the contrary, having humans fact check all news is not scalable. Thus, in this paper, we propose to approach this problem *interactively*, where humans can interact to help an automated system learn a better social media representation quality. On real world events, our experiments show performance improvements in detecting factuality of news sources, even after few human interactions.

1 Introduction

Over the last decade, we have witnessed a rise in the proliferation of "fake news" (Lazer et al., 2018), news content which lacks the journalistic standards ensuring its quality while maintaining its appearance. Social media is flooded with inaccurate and incomplete information (Vosoughi et al., 2018), and combating this has attracted significant research interest (Nguyen et al., 2020). However, this is still a hard task, particularly on unseen topics. In this paper, rather than annotating data to learn these topics, we propose to use quick **human interactions** to characterize social media, allowing us to learn a better representation, and detect factuality better.

Instead of fact checking individual articles, some works (Baly et al., 2020) focus on fact-checking sources. While still requiring automated systems due to the number of sources online, source factuality detection can be more scalable, as sources often publish content of similar factuality. Following this, we focus on capturing the factuality levels of sources: *high, mixed, low*.

One concept underlying methods that aim to exploit social information for identifying the factuality of news sources is the *social homophily principle* (McPherson et al., 2001), which captures the tendency of members of the same social group to hold similar views and content preferences. This often leads to the formation of *"echo chambers"* (Jamieson and Cappella, 2008; Quattrociocchi et al., 2016), tightly-knit *"information communities"* that have little interaction with other communities holding different views. Prior work shows how similar news, particularly misinformation, tends to spread more in some of these tightly-knit information communities (Bessi et al., 2016). Thus, identifying them can provide the needed information for capturing the factuality of sources (communities spreading mostly low factuality content in the past are likely to spread low factuality content in the future). In this work, we first capture social information in an information graph, modeling it via a R-GCN (Schlichtkrull et al., 2018).

Many approaches to detect news factuality are often studied in unrealistic settings, as their success hinges on test data being similar to or related to training data. However, a more realistic setup would examine whether a system would be able to generalize to emerging news events: These events introduce different narratives, users, and news sources, that are unseen and do not interact with training content; i.e. test users don't follow train users and test graph nodes aren't connected to train nodes. In this paper, to simulate these settings, we collected new data, consisting of the articles published around specific news events (Black Lives Matter and Climate Change - see Sec 5.1), their sources, and social context. We applied a recent strong baseline system (Mehta et al., 2022), trained over data sampled from past events, and it resulted in significant degradation in performance on the new events (∼22% Acc, 19% Macro-F1).

Our main observation in this paper is that even

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 291–311

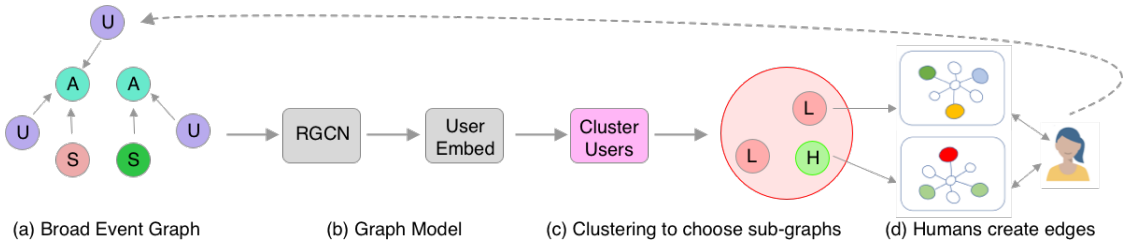

Figure 1: Our framework overview: **Adapting News Source Factuality Detection to Emerging News Events by Interactively Characterizing the Social Media Context of News Articles and Their Sources**. (Key: U = Users, A = Articles, S = Sources, Green/H = High Factuality, Red/L = Low Factuality). From the learned graph model (b), we find pairs of inconsistent users by clustering all user embeddings and looking for conflicting factuality labels (c) (L = low; H = high factuality). Here, the High Factuality user doesn't match the mostly Low Factuality cluster. We then build sub-graphs from these pairs of mismatched users and their community to show human interactors (d), who create new edges based on content similarity. This is far simpler and quicker than identifying factuality, as humans only need to identify which nodes are similar in content. Based on the interactions, we create edges in the broad event graph (a) to do better news source factuality detection (either directly or with more training).

in these challenging settings, the social homophily principle can be exploited to better detect source factuality, **if the system can identify relevant information communities over users engaging with the new content**. This is since users that are part of an information community that propagates fake news, are likely to do so as well. As we later show, automatically detecting the factuality of news sources is difficult, particularly on emerging news events. *Instead*, we suggest an **interactive learning protocol**, in which human judgements dynamically help the model identify these communities. As humans analyzing all emerging news content is clearly infeasible, we propose a novel sampling method for interactions, based on resolving inconsistencies in the model's graph-based social representation. Specifically, we identify pairs of users that are clustered in the same community, but have conflicting factuality predictions, as this indicates inconsistency. We create small sub-graphs corresponding to the social and content preferences of these users and other members of the community, and ask the humans to resolve the conflict: Based on their profile descriptions, social relations and articles endorsed, *is it likely (given the principle of social homophily) that these two users belong to the same community?* The human judgements provide rich feedback for this question, by adding edges to the graph, which connect users, articles, and sources. These edges result in cleaner information communities, which alleviate the difficulty of the source classification task. Fig. 1 describes this.

In summary, we make the following contributions. (1) We are the first to formulate the task of **interactive news source factuality detection** by characterizing social context, and implement an interaction tool for supporting this. (2) We suggest a novel sampling approach for reducing the number of human judgements needed by focusing on social inconsistencies. (3) We focus on one of the most challenging settings of news source factuality detection in emerging news events, collect data, and perform experiments showing how minimal, quick interactions can lead to performance improvements on unseen data. More generally, we propose an interactive framework to learn stronger information communities, and apply it to improve news source factuality detection. In the future, it can also be applied to other social media analysis tasks.

Sec. 3 describes our graph model, Sec. 4 our novel protocol to incorporate interactions, Sec. 5 shows results, and Sec. 6 analyzes them.

2 Related Work

Detecting fake news on social media is a popular research topic, studied in supervised learning (Hassan et al., 2017; Pérez-Rosas et al., 2018; Volkova and Jang, 2018; Ma et al., 2018; Shu et al., 2019a,b; Kim et al., 2019), Graphs (Han et al., 2020; Li et al., 2022), zero-shot (Wright et al., 2022), dialogue (Gupta et al., 2022), cross-domain (Huang et al., 2021; Zhu et al., 2022a,b; Mosallanezhad et al., 2022), and low-resource (Lin et al., 2022) settings.

One of the most challenging yet most critical social context fake news detection settings is the early detection of it, where test data has new users, articles, and sources, that do not interact with training data. Recently, researchers have been working on this task, especially at the article/tweet level. Liu and Wu classify news propagation paths, Yuan et al. model user credibility, while Konkobo et al. built a semi-supervised classifier. In our work, we focus on this challenging early detection setting, specifically to identify the factuality of news sources. We

show how our *interactive setup* can be useful, even in these settings. If combined with other early detection methods, our framework may lead to further gains, and we leave this for future work.

Using human interactions to improve models has also been popular recently (Brantley et al., 2021), in scenarios such as active learning (Blok et al., 2021), or humans providing general system feedback (Tandon et al., 2022). Other works exploit human feedback for concept discovery (Pacheco et al., 2022, 2023) by communicating human-level symbolic knowledge (Pacheco and Goldwasser, 2021). In contrast, our interactions enable stronger general models, and generalization to new unseen scenarios.

Social homophily has been used to better many NLP tasks, like sentiment analysis, entity linking, and fake news. (West et al., 2014; Yang et al., 2016; Mehta et al., 2022). Particularly, prior work shows how misinformation (and similar news) spreads more in tightly-knit communities, motivating our idea that if we use humans to increase homophily and build better information communities, we can detect facutality better (Bessi et al., 2016; Halberstam and Knight, 2016; Cinelli et al., 2021).

3 Graph Model

Similar to Mehta et al., we view fake news source detection as reasoning over relationships between sources, articles, and users in an information graph. We use their graph model[1], briefly explaining it in this sec. Sec. 4 explains our interactive protocol.

The model uses a heterogeneous graph to capture the interaction between social information and news content, and a Relational Graph Convolutional Network (R-GCN) to encode it. The R-GCN allows us to create contextualized node representations for factuality prediction. For example, one way sources are represented is by the articles they publish (which in turn are also represented by their relationships to other nodes).

Graph Creation: The graph (see: Fig.1a) consists of 3 types of nodes, each with feature vectors (details: App. A.3.1): (1) S, the news *sources*, are our classification targets. (2) A, the *articles* published by these sources, (3) U, the Twitter *users*. Sources are first connected to articles they publish. Social context is added via Twitter users that interact/connect to sources/ articles/other users. These users provide the means for fake (and real) news spread on social media: **(1) Following Sources/Users:** Users are connected to sources and users they follow. **(2) Propagating Articles:** Articles are connected to users that tweet its title/link.

Graph Embedding: As in Mehta et al., we train a R-GCN (Schlichtkrull et al., 2018) to learn graph embeddings, which will be later used to determine where human interaction may be beneficial. We optimize the Classification objective of News Source Factuality Detection (categorical cross-entropy). To predict labels, we pass the source node embeddings from R-GCN through the Softmax activation.

4 Interactive Protocol

We hypothesize that understanding content and the context it is provided in is critical to detecting fake news. Specifically, identifying information communities of users, sources, and articles based on their content preferences can be helpful, as a community that mostly shares fake news in the past, is likely to share fake news in the future. Further, users that join this community are likely to share beliefs of the community, and thus also share fake news.

Unfortunately, understanding content on social media and using it to identify information communities is challenging for AI agents. It becomes more difficult as new events with new relationships arise, as the agent does not have enough data to determine what is fake news. This makes the early detection of fake news difficult (see Sec. 5.3). On the other hand, educated humans can more easily understand relationships on social media, even in new events, as they can better analyze social interactions. Thus, humans can clear up model confusion by helping the model identify the information communities or make existing ones bigger. For example, after reading a sample of tweets from users discussing a new event, humans can *quickly* determine if the users are offering the same perspectives, and should be in the same community. This knowledge can help the agent model these users and other content they interact with better. As we later show experimentally, human interactions like these enables us to build strong information communities, which helps the agent, particularly with the early detection of news sources factuality on new news events.

Unlike automated agents, humans cannot analyze all content that pertains to a new event, as it is too massive. Instead, due to the highly connected

[1] https://github.com/hockeybro12/FakeNews_Inference_Operators

structure of social media, **small amounts of interactions done in the right places can make significant impact**, as the added information can flow throughout the information graph. Thus, we first discuss in Sec 4.1 how we determine what content humans should interact with and what interactions they should make (i.e. forming/strengthening information communities). Then, in Sec 4.2, we explain how we can incorporate those interactions back into the model to achieve performance improvements.

4.1 Soliciting Human Interactions

Now, we discuss 3 different protocols to identify the data on which humans should interact, and then what they should do. In general, we want humans to analyze a sub-graph of the broad information graph characterizing the new event. Given this sub-graph, we ask humans to help form information communities by characterizing the content in the graph based on similarity, i.e. identify if two users are similar, two articles offer the same perspective, etc. This is done by asking humans a series of questions (details : App. B) which enables them to connect nodes in the sub-graph based on content preferences, via a graphical interface we developed. An ex. is shown in App. Fig 2. We then replicate these connections in the broad information graph.

Identifying the sub-graph that will benefit the most from interactions is critical to getting the most value out of each interaction. We build the sub-graphs by first choosing a pair of users, as our end goal is to build stronger user information communities. We explore three different protocols for doing this in 4.1.1 and Sec 4.1.2. After finding these pairs of users, we build the sub-graph to show humans by including these users and their direct connections in the graph. This includes the articles they propagate, other users that propagate those articles, the sources that publish those articles, and up to 3 "influencers" (users with over 1000 followers) that one of these users follows. For each node in the sub-graph, we populate it with relevant information to enable the human interactors to understand content. For ex:, user/source nodes show user bio, tweets, etc. Article nodes show article publish date, headline, and first paragraph. Details: App B.

4.1.1 Baselines

We have two baselines for selecting pairs of users. **(1) Random**, users at random. **(2) Model Confusion** takes an *active learning*-like approach, and chooses users based on a label confusion criterion, calculated by propagating the softmax score of the source prediction downwards to get user confusion. Specifically, to get this score, we look at all the sources the user directly interacts with (articles they propagate and sources they follow), and then take the weighted average of those source's Softmax predicted label to be the user score (thus approximating user confidence). For example, a user interacting with 3 articles predicted with low factuality score of 0.7 and 1 source with high factuality score 0.9 will have confidence 0.75.

4.1.2 Social vs. Factuality Mismatch Criterion

Now, we discuss our novel protocol to determine the pairs of users, seen also in Alg 1. It is designed around one of the key ideas in this paper, homophily, the tendency of users with similar social preferences to have similar content preferences. Our graph model learns to represent both, by creating node embeddings which capture users' similarities, and learning classifiers used for characterizing content by identifying factuality. Intuitively, our protocol is designed to identify users, that based on the current model parameters, break the homophily principle. These users are part of the same social group while at the same time have different factuality predictions, and thus likely different content preferences. When this is true, the model may not have clearly understood the content preferences of these users, which a human can help clear up.

To identify these pairs of users, we first need to compute factuality labels for each user. As the model is trained for source classification, we designed a heuristic to use source labels to compute user labels: We assign users the label of the *most common predicted label* of the sources/articles the user is directly connected to. For ex:, a user following 3 low factuality sources and tweeting 1 mixed factuality article is assigned a low factuality label, as it interacts with more low factuality content.

After computing user labels, we need to find groups of similar users, which we do by k-means clustering all users in the event graph using their model embeddings (Alg 1: 3). Then, we assign each cluster a factuality label based on the most frequently occurring user factuality label in that cluster (Alg 1: 5). Finally, we choose pairs of users that are in the same cluster, but one has a different label than the cluster label, as the model thinks they are similar but predicts their factuality differently, which indicates a sign of confusion (Alg 1: 7-9).

Algorithm 1 *Social vs. Factuality Mismatch*

Input: U (Users), U_E (Graph User Embeddings), F (User Factuality Scores), P (Empty List)
Output: P (Pairs of Users To Build Graphs)
$c_{1...k} = \text{k-means}(U_E)$ `K-means Cluster all Users based on Graph User Embeddings`
for all $i = 1, \ldots, k$ **do** {for each cluster}
  $c_l = \max_{0 \leq u \leq n} F_u$ `Assign Cluster the Label of the most common user`
  for all $j = 1, \ldots, n$ **do** {for each user}
    if $F_j \neq c_l$ **then** {If user label $\neq$ cluster label}
      $U_k = \text{rand}(U)$ where $(F_j \neq F_k) \wedge (F_k \in c_l)$ `Choose a random user in the cluster with a different label`
      $P.\text{insert}((U_j, U_k))$ `Add User Pair to List`
    end if
  end for
end for
return P (Pairs of Users)

4.2 Incorporating Human Interactions

Humans interact by making new connections on the sub-graphs. We then utilize the interactions by connecting the appropriate nodes in the broader event graph. Our goal is to show how human interactions allow us to have a better model that performs well with and without further interactions.

We focus on the challenging **fully inductive setting**: where all test set nodes are not seen at training and are also not connected to training set nodes. Further, we evaluate the important setting of early detection of fake news, where test data comes from unseen emerging events. As we show in Sec 5.3, in these settings, existing models struggle.

We evaluate 3 interaction-based protocols. The three protocols have the same starting point, a graph-based factuality classification system trained over an established dataset (Baly et al., 2020). The protocols are designed to show how interactions can enhance that initial system when making predictions on data from unseen emerging events, and are organized in order of increasing effort required and increasing performance. All involve performing interactions on up to two different data sets (each corresponding to a different emerging event, see Sec 5.1). Since some of the protocols we introduce update the parameters of the model after interaction, we collect data for two events to ensure that all protocols can be evaluated in the fully inductive settings on the second event data (i.e., relying on interactions alone without training).

We hereby refer to the first event as $E1$, and the second as $E2$. Each event is further split into interaction and no interaction halves (ex: $E1$-1/$E1$-2), for comparison and model training (see below).

(1) Fully Inductive: In the first protocol, humans interact on the interaction halves of $E1$ and $E2$, and then the interactions are incorporated, without any additional training. This is the most challenging, but no extra effort is necessary for performance improvements.

(2) Interactions Amplify Model Learning: Here, our goal is to show how interactions can help us learn a stronger model that performs well without interactions. Thus, we interact on the interaction half of $E1$ (half so we can evaluate how we do on the same event without interactions), use it to train the model, and evaluate it on $E2$ (future event, fully inductive) without any additional interactions.

(3) Learning to Incorporate Interactions: In this protocol, we show how training the model after interactions allows the model to learn how to better incorporate them. This enables it to do even better when interactions are provided on future events. To do this, as above, we interact on half of $E1$ and train on it. Then, we evaluate $E2$ on both the interaction and non-interaction half. Both halves of $E2$ are connected, so although interactions are only on half, information can propagate via the graph.

4.3 Simulating Human Interactions

Due to constraints involved with human interaction time/cost, to evaluate our models we also designed a heuristic to simulate humans: We hypothesize that two users are similar if they have the same gold factuality label. While our interaction approach prioritizes content preferences for interactions, identifying this automatically is difficult, so this is an approximation. Thus, doing human interactions at the scale of simulated ones could perform better, and we leave it for future work. To get user gold factuality labels, we use the same heuristic as in Sec 4.1 (assinging users the label of the source they are most often connected to). Note that in this simulated interaction setting, we are using the test set to determine user labels, so this setup is not realistic.

5 Experiments

5.1 Dataset and Collection

To evaluate our model's ability to predict the factuality of news medium, we used the Media Bias/Fact Check dataset (Baly et al., 2018). We expand it by scraping additional sources from Media Bias/Fact Check[2], for better coverage of recent events and increasing the number of sources for evaluation.

[2]https://mediabiasfactcheck.com

Model	Baly Acc.	Baly F1	E1-1 Acc	E1-1 F1
Baly	71.52	67.25	-	-
Mehta R-GCN	68.90	63.72	-	-
Mehta BEST	72.55	66.89	-	-
BL: Mehta R-GCN	66.04	54.20	43.21	34.44

Table 1: Baseline results on Baly (Baly et al., 2020) and an inductive future BLM event $E1$-1 (not seen or connected to the training graph). Baseline (BL) is the strong graph classification model from (Mehta et al., 2022) (Mehta R-GCN) that was competitive with the state of the art ((Mehta et al., 2022) - Mehta BEST). Even with this, performance significantly worsens on $E1$, showing that detecting fake news on future events inductively is challenging. BL: Mehta R-GCN was trained on a smaller (Baly et al., 2020) dataset, as some sources were used for evaluation, which is why the performance is slightly lower.

Identically to Baly et al., we labeled the sources on a 3-point factuality scale: *high*, *mixed*, or *low*.

Our goal in this paper is to show how human interactions can help news source factuality detection on new events, where even strong models struggle (Sec 5.3). To do this, we evaluated our model on **two broad events**: *Black Lives Matter (BLM)* and *Climate Change (CLM)*. For each, we scraped data from Twitter over 3 time periods (01/02/2019 - 06/01/19; 06/02/19 - 01/1/21; 02/02/21 - 05/06/22), each of which additionally cover many different sub-events. For each time period, we created a fully inductive graph, consisting of at least 99 sources and their metadata. None of these graphs are connected to each other in any way and no nodes in any of them are common with each other or the training set - making our test settings fully inductive, and very challenging. To ensure this inductive setting, when collecting data for future time periods, we made sure not to include sources/users/articles that we already used in previous time periods, even if they propagated content in those future periods. Combined with Baly et al., we used the first time period for training, to teach the model how to identify fake news in general and as it pertains to an event. We used the 2nd and 3rd time period as $E1$ and $E2$ in the protocols discussed in Sec 4.2. Details (statistics, etc.): A.2. We release our code and data.[3]

5.2 Evaluation Method

For both BLM and CLM, we evaluate on the two inductive sub-events ($E1, E2$) collected in Sec 5.1. The interaction half is referred to with a -1 and the non-interaction with -2, for ex: $E1$-1. For fair comparison, each data split, i.e. $E1$-1, is the same across all evaluations. We report results on Accuracy, Macro F1 (the dataset is unbalanced), and the total number of edges added by all interactions.

[3] https://github.com/hockeybro12/Fake_News_Interactive_Detection

Model	E2-1 Acc	E2-1 F1
Random Users	35.21	29.99
Confused Users	36.61	**32.72**
User Clustering	**42.10**	32.22

Table 2: Ablation study on our methods for choosing interactions on $E2$-1. It is clear that finding users based on clustering and then factuality mismatch is best.

We evaluated 3 settings, the first 2 are simulated using gold test set labels (see Sec 4.3), while the last is done by humans: **(1) Interaction Graphs Only:** Edges are added only between users in interaction graphs. **(2) X% of Data:** Edges are added between X% of all possible users that have the same label in the test set that we run interactions on (not only users in interaction sub-graphs). X is 100%, 75%, or 25%. **(3) Human Interactions**: For BLM, we evaluate on two separate versions, each featuring a different source set (and social media data). This section shows results on the first version, when one human interacts on 20 graphs per data split (details in B.1, B.2). The appendix shows the second version of BLM with ***3 different interactors***, showing the same trends. For space, these interaction results/details (including agreement) are in B.2.3. This section also evaluates Climate Change, with 2 interactors interacting on 10 sub-graphs per data split. (for detailed CLM results, see App. C). The human interaction results are also the most realistic evaluation setting, as they don't use any gold test set labels, like the simulated interactions do.

5.3 Baselines

We trained our baseline model, from Sec. 3, for Source Factuality Detection on Baly et al. and the first event, where it achieved strong performance, similar to SOTA (Mehta et al., 2022) (we use the same data and methodology) and other baselines (Baly et al., 2020) (SVM). However, when evaluated inductively on a BLM event that was published after dates the training data was collected from - i.e. $E1$-1 - performance significantly worsened (see Table 1). This validated our hypothesis that strong models, even if trained on generic and event specific data, do not translate well to future events. Thus, we propose to use our interactive protocol.

Model	E1-1 Acc	E1-1 F1	E1-2 Acc	E1-2 F1	E2-1 Acc	E2-1 F1	# Edges
BLM No Interactions	43.21	34.44	37.93	30.70	35.21	27.65	-
CLM No Interactions	40.16	32.77	39.65	31.86	34.88	30.93	-
BLM Sim. Interactions on Sub-Graphs Only	44.54	36.45	37.93	30.70	42.10	32.22	2,162
BLM Sim. Interactions on 100% of Data in E1-1 + E2-1	49.20	40.52	37.93	30.70	44.73	36.82	133,336
BLM Sim. Interactions on 75% of Data in E1-1 + E2-1	46.03	38.05	37.93	30.70	50.00	40.50	74,414
BLM Sim. Interactions on 25% of Data in E1-1 + E1-2	46.03	37.65	37.93	30.70	42.10	32.65	8,266
BLM Human Interactions in E1-1 + E2-1	44.44	35.96	37.93	30.70	44.73	30.03	84
CLM Human Interactions in E1-1 + E2-1	46.72	43.94	39.65	31.86	39.53	36.95	47

Table 3: Protocol 1: Interactions results on BLM and Climate Change (CLM) in the difficult, inductive, no training setting. E1 and E2 are the two separate, inductive graphs. E1-1 is the first half that receives interactions, and E1-2 is the second half that doesn't. E2-1 (first half E2) also receives interactions, but it's dev set is not used to select the model. With a minimal number of added edges, human interactions achieve performance improvements in these difficult, inductive settings, with no extra training (compared to No Interactions). Ex: results improve on human BLM E2-1 (∼9.5% Acc.) Sim. settings also show improvements.

Model	E1-2 Acc	E1-2 F1	E2-1 Acc	E2-1 F1	# Edges
BLM No Interactions	37.93	30.70	35.21	27.65	-
BLM No Interactions Train	64.86	66.91	42.10	40.10	-
CLM No Interactions Train	49.29	44.84	44.77	42.35	-
BLM Sim. Interactions on Sub-Graphs Only	62.16	62.95	43.66	29.47	2,162
BLM Sim. Interactions on 100% of Data in E1-1	56.75	59.27	45.07	40.18	133,336
BLM Sim. Interactions on 75% of Data in E1-1	65.51	64.01	43.66	39.11	74,414
BLM Sim. Interactions on 25% of Data in E1-1	54.05	46.48	39.43	35.09	8,266
BLM Human Interactions in E1-1	67.56	71.56	45.07	35.18	84
CLM Human Interactions in E1-1	53.52	44.53	40.29	46.38	47

Table 4: Protocol 2: Interactions results on BLM + Climate Change (CLM) when we train on interactions, and then apply the model to a new event with no interactions done. E1 and E2 are the two separate, inductive graphs. E1-1 is the interaction half of the 1^{st} event and E1-2 is the 2^{nd}, non-interaction half. E2-1 (non-interaction half) is not connected to E1. Compared to the model that was trained on E1-1 without interactions (No Interactions Train), human interactions lead to a more accurate model for future events, by ∼3% better Acc. for BLM and ∼4% F1 for CLM (E2-1). Sim. settings also show improvements.

5.4 Interactions

We now evaluate our interaction protocols - what portions of the graph to show users and how to incorporate interactions, using the method in Sec 5.2.

5.4.1 Soliciting Interactions

When comparing our methods for choosing what sub-graphs to show on BLM, simulated interactions performance shows a benefit (Table 2) of choosing the users to build interaction graphs for based on confused user clustering. This matches our intuition as if the model predicts a users' factuality differently than other users similar to it, then the model is confused and clearing that could improve performance. Thus, we use this method of choosing sub-graphs throughout the rest of our experiments.

5.4.2 Incorporating Interactions

Now, we evaluate our 3 protocols of incorporating interactions discussed in Sec 4.2, in order of increasing performance and model training required. For space, additional human interaction results are in App. B.2.3 and detailed CLM results in App. C. Note that simulated interactions (4.3) use gold test set labels and thus are only used to test our models.

First, Protocol 1, where we evaluate how the model performs with interactions in the completely inductive setting, so no training is necessary. In Tab. 3, we ran interactions on only the interaction half of each event ($E1$-1 + $E2$-1) and the dev. data (also from $E1$-1), to choose the strongest model. To ensure the dev. set being chosen from $E1$-1 does not bias us into a strong model, we also did interactions on the interaction half of $E2$ and notice stronger performance improvements. Note that $E2$ is a future event and is not connected to $E1$ at all. All settings improve performance. Moreover, on BLM, human interactions improves performance ∼9.3% Acc. on E2-1, comparable to simulated interactions with significantly more data, showing the large impact benefit of human interactions.

Next, in Protocol 2, we learn a better model for

Model	E1-2 Acc	E1-2 F1	E2-1 Acc	E2-1 F1	E2-2 Acc	E2-2 F1	# Edges
BLM No Interactions	37.93	30.70	35.21	27.65	30.30	24.84	-
BLM No Interactions Train	64.86	66.91	42.10	40.10	45.45	42.35	-
CLM No Interactions Train	49.29	44.84	44.77	42.35	44.44	33.06	-
BLM Sim. Interactions on Sub-Graphs Only	62.16	62.95	57.89	48.53	45.45	43.49	2,162
BLM Sim. Interactions on 100% of Data in E1-1 + E1-2	56.75	59.27	57.89	61.90	36.36	35.53	133,336
BLM Sim. Interactions on 75% of Data in E1-1 + E1-2	65.51	64.01	63.15	61.84	45.45	43.60	74,414
BLM Sim. Interactions on 25% of Data in E1-1 + E1-2	54.05	46.48	44.73	31.38	51.51	45.31	8,266
BLM Human Interactions in E1-1 + E2-1	67.56	71.56	50.00	43.60	51.51	40.09	84
CLM Human Interactions in E1-1 + E2-1	53.52	44.53	53.48	43.07	46.80	38.73	47

Table 5: Protocol 3: Results on BLM + Climate Change (CLM) when we train on interactions and then do more in the inductive setting. E1 and E2 are the two separate inductive graphs. E1-1 is the interaction half of E1 that is trained on. E1-2 is the non-interaction half. E2 receives interactions on the interaction half (E2-1), but not the non-interaction half (E2-2). Human interactions improve accuracy on both halves of E2 and F1 on E2-1, compared to no interactions train, and more than only applying interactions without training for them as Tab.3, showing the benefit of training to learn to incorporate interactions.

news source factuality detection after doing interactions, compared to not doing any. In Tab. 4, we ran interactions on the interaction half of $E1$, and then trained on that data. On $E2$ with no interactions done, we can see how this improves accuracy compared to models trained without interactions.

Finally, for Protocol 3, we learn to better incorporate interactions into the model after we train for it. Thus, we train similarly to Protocol 2, but now we also run interactions on the interaction half of $E2$. In Tab. 5, we see accuracy improves on both halves of $E2$ after we learn to incorporate interactions on $E1$, even though $E2$ is inductive. Further, F1 improves on $E1$-1. This shows that training with and then doing interactions helps performance significantly on future events. We hypothesize that this happens as training with interactions enables the model to learn how to incorporate them better, allowing the model to further take advantage of them whenever provided. Further, human interactions based on content preferences provide clearer results compared to simulated ones (without cheating and using test set labels), as the model better learns the social media landscape, shown by it achieving better accuracy on the BLM interacted and non-interacted data (both halves of $E2$).

From these results, we see that our real-world applicable human interactions models result in performance improvements in either Accuracy or Macro F1, often times both. As a whole, all our models improve performance (any non-gain in one of these metrics is offset by significant gains in the other). We additionally hypothesize that performing more interactions (particularly human) will achieve higher and more consistent results.

Model	Purity	# Edge
No Inter.	36.2, 37.8, 33.3	-
P1: Inductive Human	39.2, 40.2, 35.3	84
P2: Train Human	49.5, 37.4, 41.4	84
P3: Train + Inter. Human	53.4, 41.9, 42.6	84

Table 6: Purity clustering (sources, articles, users) for the human interaction protocols on E2-1. As training increases with each protocol (P), purity does too, showing that interactions do help to learn better information communities.

6 Discussion

Now, we analyze our best BLM interaction model for fake news source detection (for each protocol) on $E2$-1 by answering these research questions:
(1) *Do interactions help learn better communities?*
(2) *What pairs of nodes do humans connect?*
(3) *How can our model be used in the real world?*
(4) *Do interactions change embeddings?* App. D.1

6.1 Learned Communities

We analyze how interactions help learn better info. communities. We evaluate cluster-purity by K-means clustering sources, articles, and users before and after interactions are done. To compute purity, each cluster is assigned to the class which is most frequent in it, and then the accuracy of this is measured. Users are assigned gold labels based on the most common label of all the nodes they are directly connected to in the graph. Results in Tab. 6 show purity increases after interactions, showing interactions help learn better communities.

6.2 Human Interaction Analysis/Examples

We analyze the interactions to determine what humans connected. We see humans make smart decisions in matching content preferences. Further, we show specific examples, demonstrating the ease,

quickness, and lack of subjectivity of the interaction process. These details/ex. are in App. D.2.

6.3 Real World Use Case

As shown in Sec. 5, our interactive protocols enable rapidly (humans spent ∼3 min/sub-graph) learning better source factuality detection models for new events, even in the most challenging settings when there are no users, articles, or sources in common with prior data. This happens as contrary to providing additional labels, which can be time consuming and hard, interactions clear up content preferences, creating better social homophily and performance.

Specifically, in a real-world use case, interacting at training time learns a better model for the new event setting (Protocol 2 results on E2-1). In addition, this model would become even stronger as more interactions are performed, even without any further training, as seen in Protocol 3. Thus, when new news events happen, humans can interact on a few settings (our interaction sub-graphs) and our setup enables the model to amplify this knowledge to rapidly detect fake news sources on a large scale.

7 Summary and Future Work

We proposed an initial protocol to interactively build stronger information communities, applying it on source factuality detection. We focused on the early detection settings, where even strong models can struggle. Our approach of finding sub-graphs and then interacting on them via 3 protocols enables minimal, quick human interactions to achieve significant performance improvements. We hypothesize that our interactive framework can generalize to other social media analysis tasks like bias or topic detection, and testing it is our future work. Additionally, we aim to scale up our interaction process, to include additional human interactions and types of interactions.

8 Acknowledgements

We thank the anonymous reviewers of this paper for all of their vital feedback. The project was funded by NSF CAREER award IIS-2048001 and IIS-2135573.

9 Ethics Statement

In this section, we first discuss some limitations of our model (9.1), and then expand on that with a discussion on ethics as it relates to our data collection, data usage, human interaction, and the deployment of our models (9.2).

9.1 Limitations

This work tackles fake news source detection in English on Twitter (our social media platform of choice). Our methods may or may not apply to other languages with different morphology, and other social networking platforms. We leave the investigation of this to future work, but are optimistic that especially with the benefit of interactivity, our methods may generalize.

The nature of our interactive framework also requires human interactors to interact, which could be a potential limitation. Interactors must have some general understanding of news content and be able to identify if two entities (users, sources, or articles) have similar content relationships. However, as interactors are just looking for content/perspective similarity, they need not be aware of the latest events or be fake news detection specialists. Further, human interactors don't analyze user-specific information or profile users themselves, they just determine if users have similar content relationships.

We used a single GeForce GTX 1080 NVIDIA GPU to train our models, with 12 GB of memory. As our models are largely textual based, they do not require much GPU usage. However, scaling our experiments to larger scale settings in real world settings could require more compute, which may be a potential limitation. Our hyper-parameter search, mentioned in App A.3 was done manually.

9.2 Ethics

To the best of our knowledge no code of ethics was violated throughout the experiments done in this paper. We reported all hyper-parameters and other technical details necessary to reproduce our results, and release the code and dataset we collected. We evaluated our model on two datasets that we collected in this paper, and was collected by prior work, but it is possible that results may differ on other datasets. We believe our methodology is solid and applies to any social media fake news setting. Due to lack of space, we placed some of the technical details and discussion to the Appendix section. The results we reported supports our claims in this paper and we believe it is reproducible. Any qualitative result we report is an outcome from a machine learning model that does not represent the authors' personal views. For

anything associated with the data we use, we do not include account information and all results are anonymous.

In our dataset release, we include sources, users, and articles, with enough data to produce the results described in the paper and the Appendix. Sources are public information provided in (Baly et al., 2020), and we map each to an ID. We release article graph embeddings, which can be used to train our models. As these embeddings are neural network representations, they can't be mapped back to article text. However, we also release article URLs, so that the articles can be downloaded, if they are still publicly available. Additionally, we release the Twitter data that we used, in complicance with the Twitter API policies [4]. In our dataset release, each user is referenced by their Twitter ID, and their graph ID (the graph ID is meaningless on it's own). We release the mapping of the Twitter ID to the graph ID. By us only releasing Twitter ID's, and not the actual Twitter text or user information, in order to download the exact Twitter data that we used, users must use the Twitter API to gather the latest public information[5]. This ensures that we respect user privacy, in accordance with the policies mentioned by the Twitter API, as only user content that is still public can be downloaded and we are not storing/releasing any data. We also provide the model representations for each user, article, and source we used as our initial embedding in the graph. As these are neural network model embeddings, they can't be mapped back to the individual text. Our data is meant for academic research purposes and should not be used outside of academic research contexts. All our data is in English.

In this paper, we did not use any of the Twitter data for user surveillance purposes, and we encourage the community to do the same, to respect user privacy. We also do not profile users, we only use the user insights as an aggregate to classify news sources. Further, we only use public Twitter profiles, which there are enough of for our framework to work in real-time situations. When doing human interactions, we show humans public Twitter information, so that they can determine user similarity. To do this, we use the Twitter API to determine the Twitter data that is publicly available at the time of interaction, show that to humans, and then discard the Twitter information. Further, in our graph model, we do not store any user-specific information, we only store neural network model embeddings which are used for training and cannot be mapped back to the original text or user. The same is true for articles, so we we are actually discarding all the text (Twitter, article, and source). Users of our framework should also do the same - use public knowledge for interactions and not store any user/article specific data, rather use the appropriate APIs to retrieve the data when needed.

Our framework in general is intended to be used to defend against fake news. While our framework could be used to build better methods of designing fake news, our methodology of interactive fake news detection could guard against that as well. We caution that our models and methods be considered and used carefully. This is because in an area like fake news detection, there are great consequences of wrong model decisions, such as unfair censorship and other social related issues. Further, despite our efforts, it is possible our models are biased, and this should also be taken into consideration. Our protocol of building sub-graphs based on model confusion that we used when showing humans what to interact on, can be used to get insights into the model to help prevent some of these issues as well. However, this is definitely an area of future work.

In the interactive setting we proposed, our approach relies on getting insights from human interactors and using that to improve performance in fake news detection. While that lead to performance improvements in this work and we believe it will hold in different settings, there could be issues, such as biased humans. Running interactions at large scale with multiple human experts per subgraph can help mitigate some of these issues. For example, edges can be weighted in the graph based on how many humans chose to add them. Thus, extremely biased interactors decisions would be given less weight, and maybe even not considered by the model. We leave this for future work. However, despite this, there may still be some human interactor bias that can leak into the final fake news detection model, which is why perhaps important decisions should not be made only by machine learning models, but rather the models be used as a tool.

As mentioned in the Appendix B.2.3, the human interactors we used were Compute Science PhD

[4] https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases

[5] https://developer.twitter.com/en/docs/twitter-api

students. The interactors were awarded research credits for their work, as the hours they spent working on the task were considered as part of their research credit hours. They were explained the entire process before hand including what the interactions would be used for, and agreed to perform the interaction. The total interaction process took under 3 hours, including the time spent explaining the process.

These and many other issues are things to consider when using fake news detection models such as the one proposed in this work.

References

Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. 2018. Predicting factuality of reporting and bias of news media sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, Brussels, Belgium.

Ramy Baly, Georgi Karadzhov, Jisun An, Haewoon Kwak, Yoan Dinkov, Ahmed Ali, James Glass, and Preslav Nakov. 2020. What was written vs. who read it: News media profiling using text analysis and social media context. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20.

Alessandro Bessi, Fabio Petroni, Michela Del Vicario, Fabiana Zollo, Aris Anagnostopoulos, Antonio Scala, Guido Caldarelli, and Walter Quattrociocchi. 2016. Homophily and polarization in the age of misinformation. *The European Physical Journal Special Topics*, 225:2047–2059.

Pieter M Blok, Gert Kootstra, Hakim Elchaoui Elghor, Boubacar Diallo, Frits K van Evert, and Eldert J van Henten. 2021. Active learning with maskal reduces annotation effort for training mask r-cnn. *arXiv preprint arXiv:2112.06586*.

Kianté Brantley, Soham Dan, Iryna Gurevych, Ji-Ung Lee, Filip Radlinski, Hinrich Schütze, Edwin Simpson, and Lili Yu, editors. 2021. *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*. Association for Computational Linguistics, Online.

Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. 2021. The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9):e2023301118.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Dialfact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801.

Yosh Halberstam and Brian Knight. 2016. Homophily, group size, and the diffusion of political information in social networks: Evidence from twitter. *Journal of public economics*, 143:73–88.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Yi Han, Shanika Karunasekera, and Christopher Leckie. 2020. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812.

Yinqiu Huang, Min Gao, Jia Wang, and Kai Shu. 2021. Dafd: Domain adaptation framework for fake news detection. In *International Conference on Neural Information Processing*, pages 305–316. Springer.

Kathleen Hall Jamieson and Joseph N Cappella. 2008. *Echo chamber: Rush Limbaugh and the conservative media establishment*. Oxford University Press.

Jooyeon Kim, Dongkwan Kim, and Alice Oh. 2019. Homogeneity-based transmissive process to model true and false news in social networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 348–356.

Pakindessama M Konkobo, Rui Zhang, Siyuan Huang, Toussida T Minoungou, Jose A Ouedraogo, and Lin Li. 2020. A deep learning model for early detection of fake news on social media. In *2020 7th International Conference on Behavioural and Social Computing (BESC)*, pages 1–6. IEEE.

David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Ke Li, Bin Guo, Jiaqi Liu, Jiangtao Wang, Haoyang Ren, Fei Yi, and Zhiwen Yu. 2022. Dynamic probabilistic graphical model for progressive fake news detection on social media platform. *ACM Transactions on Intelligent Systems and Technology (TIST)*.

Hongzhan Lin, Jing Ma, Liangliang Chen, Zhiwei Yang, Mingfei Cheng, and Chen Guang. 2022. Detect rumors in microblog posts for low-resource domains via adversarial contrastive learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2543–2556.

Yang Liu and Yi-Fang Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1980–1989.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444.

Nikhil Mehta, María Leonor Pacheco, and Dan Goldwasser. 2022. Tackling fake news detection by continually improving social context representations using graph neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1363–1380.

Ahmadreza Mosallanezhad, Mansooreh Karami, Kai Shu, Michelle V Mancenido, and Huan Liu. 2022. Domain adaptive fake news detection via reinforcement learning. In *Proceedings of the ACM Web Conference 2022*, pages 3632–3640.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1165–1174.

Maria Leonor Pacheco and Dan Goldwasser. 2021. Modeling content and context with deep relational learning. *Transactions of the Association for Computational Linguistics*, 9:100–119.

Maria Leonor Pacheco, Tunazzina Islam, Monal Mahajan, Andrey Shor, Ming Yin, Lyle Ungar, and Dan Goldwasser. 2022. A holistic framework for analyzing the COVID-19 vaccine debate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5821–5839, Seattle, United States. Association for Computational Linguistics.

Maria Leonor Pacheco, Tunazzina Islam, Lyle Ungar, Ming Yin, and Dan Goldwasser. 2023. Interactive concept learning for uncovering latent themes in large text collections. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5059–5080, Toronto, Canada. Association for Computational Linguistics.

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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401.

Walter Quattrociocchi, Antonio Scala, and Cass R Sunstein. 2016. Echo chambers on facebook. *Available at SSRN 2795110*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019a. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 395–405.

Kai Shu, Deepak Mahudeswaran, and Huan Liu. 2019b. Fakenewstracker: a tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, 25(1):60–71.

Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. 2022. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. *NAACL Findings.(to appear)*.

Svitlana Volkova and Jin Yea Jang. 2018. Misleading or falsification: Inferring deceptive strategies and types in online news and social media. In *Companion Proceedings of the The Web Conference 2018*, pages 575–583.

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

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, et al. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2:297–310.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460.

Yi Yang, Ming-Wei Chang, and Jacob Eisenstein. 2016. Toward socially-infused information extraction: Embedding authors, mentions, and entities. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1452–1461.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2020. Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5444–5454.

Yongchun Zhu, Qiang Sheng, Juan Cao, Shuokai Li, Danding Wang, and Fuzhen Zhuang. 2022a. Generalizing to the future: Mitigating entity bias in fake news detection. In *Proceedings of the 45nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery.

Yongchun Zhu, Qiang Sheng, Juan Cao, Qiong Nan, Kai Shu, Minghui Wu, Jindong Wang, and Fuzhen Zhuang. 2022b. Memory-guided multi-view multi-domain fake news detection. *IEEE Transactions on Knowledge and Data Engineering*.

A Supplemental Material: Fake News Source Detection

In this section, we provide implementation details for our models for fake news source detection. The original dataset we used has 859 sources: 452 *high* factuality, 245 *mixed*, and 162 *low*, and was released publicly by (Baly et al., 2020)[6]. We then extended it by scraping sources from the Media Bias/Fact Check website[7], to gain better coverage of more recent events. The dataset does not include any other raw data (articles, sources, etc.), so we must scrape our own.

A.1 Data Collection

We will release the data and code for this paper upon acceptance. Our data collection process is identical to Mehta et al. as we use their code, but we briefly describe the process here completion. Further details are available in Mehta et al.

Following the process used in Mehta et al., we tried to scrape news articles for each source in the dataset using public libraries (Newspaper3K [8], news-please [9] (Hamborg et al., 2017), and Scrapy [10]). In cases where the webpage of the news source was removed as often happens with fake news websites, we used the Wayback Machine [11] to download the articles, if possible. As explained in Sec 3, we attempted to get up to 300 articles for each source. For the sources we got for the (Baly et al., 2020) dataset, our statistics are the same as Mehta et al. as we use their data, and thus the sources have an average of 109 articles with a STD of 36.

To scrape Twitter data, we used the Twitter API[12]. In order to densely populate the graph, we attempted to scrape up to 5000 followers for each source that had a Twitter account ((72.5% of the sources, the same number as (Baly et al., 2020; Mehta et al., 2022)). Further, to find users that propagate articles, we used the Twitter Search API to search articles. From the returned Twitter results, we use users that mention the article title or the article URL and post their tweet within 3 months of the original article being published. For each user found, we download their profile and add them to our graph, making the appropriate Twitter User to

[6]https://github.com/ramybaly/News-Media-Reliability
[7]https://mediabiasfactcheck.com
[8]https://github.com/codelucas/newspaper
[9]https://github.com/fhamborg/news-please
[10]https://github.com/scrapy/scrapy
[11]https://archive.org/web/
[12]https://developer.twitter.com/en/docs

Article connection as discussed in Sec 3. Finally, we scraped the followers of each Twitter user in the graph, and connected them to any user they followed that was in the graph. This increases the connectivity of the graph and allows us to better capture the social media landscape. To maintain a densely connected graph, we remove users that do not connect to any other node in the graph.

To get the data for the YouTube embeddings for the sources, we used the ones released publicly by (Baly et al., 2020), who were able to scrape YouTube channels for 49% of sources.

A.2 Event Collection

To collect the data for each event ($E1$ from June 2, 2019 - Jan 1, 2021 and $E2$ from Feb 2, 2021 - May 6, 2022), we filtered by date and downloaded Tweets mentioning certain hashtags and a URL with one of the sources in the dataset. The hashtags/search terms we used to collect data for the *Black Lives Matter* event were: *Black Lives Matter, BLM, blacklivesmatter, Floyd, George Floyd*.

We then filtered the data to find the top *high*, *mixed*, and *low* factuality sources that were mentioned on Twitter for each of these events and time periods. We kept sources that had at least 10 articles with Twitter users propagating them. We ended up with at least 33 sources for each factuality level in each event data split. Other sources that were in the training set of Baly et al. were used to train the initial Graph Embedding. Detailed statistics for the training set and the sources used in one of our Black Lives Matter splits are shown in Table 8.

A.3 Experimental Settings

A.3.1 Initial Embeddings

We used the same initial graph node embedding representations as Mehta et al., which we briefly explain here. The Twitter embedding we used for each source and each Twitter user was a 773 dimensional vector consisting of the SBERT (Reimers and Gurevych, 2019) (RoBERTa (Liu et al., 2019) Base NLI model) representation of their bio (up to the first 512 tokens) concatenated with a variety of numerical features, as follows: (1) a binary number representing whether the source is verified, (2) the number of users a source follows and the number that follow it, (3) the number of tweets the user posts, and (4) the number of favorites/likes the users' tweets have received. The YouTube embedding we used consisted of the following numerical features: the average of the number of views, the number of dislikes, and the number of comments for each video the source posted on YouTube. For articles, we used the same SBERT RoBERTa model to generate an embedding based on the article text, which ends up being a 768 dimensional vector. In all cases where we encoded text for embedding representations, we use SBERT (Reimers and Gurevych, 2019) RoBERTa (Liu et al., 2019) model because it provides semantically meaningful sentence representations for the text.

A.3.2 Models and Training

We used the publicly released code of Mehta et al., which was built using PyTorch (Paszke et al., 2019) and DGL (Deep Graph Library) (Wang et al., 2019) in Python. The R-GCN used consists of 5 layers, 128 hidden units, a learning rate of 0.001, and a batch size of 128 for Node Classification. The initial source and article embeddings have hidden dimension 768, while the user one has dimension 773. To do 3-way source classification, the final fully connected layer has size 3.

We trained our models using a 12GB TITAN XP GPU card. To learn the initial model which was used to determine where to perform interactions, it took approximately 2 hours. Training after interactions takes approximately 30 minutes total. Inductive settings do not have any training, and take minutes to run as we only have to compute embeddings for nodes that we are attempting to classify.

We used the development set to evaluate model performance, and choose the best hyperparameters.

B Supplemental Material: Interaction Graphs

In this section, we provide details about the interaction graphs we showed users in B.1. Then, in B.2, we discuss the interaction process (B.2.1), the details behind the interactions used in the main paper (B.2.2), and finally new interactions on the second Black Lives Matter Split (B.2.3).

B.1 Interaction Graph Details

The sub-graphs were constructed by first picking pairs of users, and then adding context around them, as discussed in Sec 4.1. The context consists of the article each user propagated, the sources that published it, other users that propagated the same arti-

Model	T1-1 Acc	T1-1 F1	T1-2 Acc	T1-2 F1	T2-1 Acc	T2-1 F1	# Edges
No Interactions	41.79	37.10	41.93	35.95	37.50	33.54	-
No Interactions Train	85.71	85.16	68.42	63.00	40.40	30.75	-
Protocol 1: Human Interactions in T1-1	43.28	37.39	45.16	42.34	-	-	65
Protocol 2: Human Interactions in T1-1	71.68	72.50	52.63	41.82	41.41	34.79	65

Table 7: Additional interaction results across the two of our protocols. With minimal interactions, we still see performance improvements in the fully inductive setting in Protocol 1 using all the interactions done by the three humans. We also see how the interactions allow us to learn a better model in Protocol 2, as seen by the improvements on $T2$-1, despite no interactions performed there (interactions were only on $T1$-1).

Dataset	Low	Mixed	High
Training Event	57	81	153
T1	56	33	33
T2	33	33	33

Table 8: Number of sources in our datasets for the training event (added to the training set to train the initial model), T1 (first event) and T2 (second event). The results on these data sets are shown in Table 7.

cles, and finally some additional celebrities (users with more than 1000 followers) that are followed by one of the users in the graph. All articles added to the graph have to be about the same event, and they are found by searching hashtags related to the event (in our case Black Lives Matter or Climate Change, hashtags below) in a four week span on Twitter. An example of an interaction graph that is shown to humans is in Fig. 2

Each node in the interaction graphs also contains metadata to provide more information to the human interactors. As seen in Fig 3, article nodes consist of the headline, article text, article entities, and the date the article was published. As seen in Fig 4 source and user nodes contain Twitter information such as: username, following count, number of followers, whether they are verified, how many tweets they make, what is their model predicted label, their biography, the tweet they made about the article, and other tweets they made about the same event.

Once the graphs are built, we show them to humans and ask a series of questions to guide the interaction process. Each question asks the user to create an edge in the interaction graph based if there is a positive relationship between the two nodes. A Positive relationships mean the nodes have similar content preferences. If a positive relationship doesn't exist, or there is not enough data to clearly determine a positive relationship, humans are asked to not make any edge connections. Thus, humans are asked to ignore potentially subjective cases. Humans identifying positive relationships has multiple benefits for fake news detection: **(1) Simplicity:** It is far simpler than identifying factuality, so it can be used to detect fake news quickly. The simplicity of the interaction process is due to the fact that interactors only have to read a small amount of content (a few user tweets/profile information + up to two article headlines/summaries), compared to reading multiple articles and gaining real world knowledge. On average, human interactors spent 3 minutes per interaction graph, and made an average of 8 connections in this time. **(2) Effectiveness:** Interactions improve social media representation quality and thus social homophily, and that's what leads to performance improvements. Here are the questions we asked human interactors for each sub-graph shown:

1. Are there any users that are similar to each other? Please connect them.
2. Are there any articles that are similar to each other? Please connect them.
3. Are any users likely to propagate any of the articles? Please connect them to the appropriate article.
4. Are any users likely to interact with another user? Please connect those pairs of users.
5. Are any users likely to interact with any sources? Please connect those users of the respective source.

B.2 Human Interactor Details

In this sub-section, we first discuss the interaction process, including the graphical interface we built for this task (B.2.1). Then, we discuss the two rounds of human interaction protocols we did for

Black Lives Matter. Climate change details follow in Sec C. The first split of Black Lives Matter, Sec B.2.2 was presented in the main paper, and the second split B.2.3 appears in this section.

B.2.1 Interaction Process

The human interactors use a graphical interface that displays the interaction graphs. We hosted the interface on a website built for this interaction process. Humans must answer the questions discussed above (Sec B.1) by connecting nodes to create edges, which are then saved on our server and can be incorporated into the broad event graph as new edges, when evaluating the overall performance. The examples of the graphs human interactors see when interacting, and the metadata they are provided with, can be seen in Fig 2, Fig 3, and Fig 4. Examples of connections made are in 6.2.

B.2.2 Initial Interactor Details

We now describe the initial interaction session we discussed in the main Paper on Black Lives Matter and used for all the BLM Human results presented there. The interactor used for this session was an Asian-American PhD student in Computer Science and Natural Language Processing. They were explained of the entire process before hand including what the interactions would be used for, and agreed to perform the interaction. They interacted on 20 graphs per data split (E1-1, dev, and E2-1), which took under an hour for each split.

B.2.3 Additional Interactor Details

To expand our human interactor sessions, we also ran additional interaction sessions on the Black Lives Matter dataset (and Climate Change in Sec C. For variety, we used a different source split in this setting, that we will also release. For this reason, these additional results are not comparable to the ones in the main paper, and we refer to the new events as $T1$, $T1$-1, $T2$, etc. instead of $E1$, $E1$-1, $E2$, etc. The data collection was the same as before.

For these additional interactions, we used three interactors of Asian descent, all fluent in the English language and all Computer Science PhD students. The interactors were awarded research credits for their work, as the hours that they spent working on the task were considered as part of their research credit hours. They were explained of the entire process before hand including what the interactions would be used for, and agreed to perform the interaction. Each interactor interacted on 10 graphs for $T1$-1, and spent less than one hour. A majority of the time was spent becoming familiar the interaction process, and once complete the interactions went more rapidly. As a test, we had one interactor do interactions on 10 more graphs, and they were able to do 10 graphs in less than 30 minutes, showing how this process can be done rapidly. Moreover, interactors spent an average of 3 mins. per interaction sub-graph, once familiar with the process. Across the three interactors, we had 65 unique edge connections made for $T1$-1. We had 31 edges that were repeated across the interactors, showing a reasonable level of interactor agreement given the the task.

Results for this additional interaction process is in Tab. 7, and they are consistent with the results for the Protocols in the main paper for the single interactor. For Protocol 1, we see improvements in the inductive setting on both the interaction and non-interaction half of $T1$. Thus, interactions help performance even when there is no additional training. Protocol 2 also leads to improvements and shows how interactions (done on $T1$-1) allow us to learn a better model for when no interactions are done (dev. set performance was 43.45% Accuracy). This further shows how interactions can help to build a stronger model, especially on emerging news events. In addition, it is likely that more interactions would lead to more improvements, and we leave this for future work.

C Supplemental Material: Climate Change

In this section, we expand upon the Climate Change (CLM) dataset results discussed first in Sec 5. We first discuss what search terms we used to collect the data for Climate Change in Sec C.1. Then, in Sec C.2, we explain the interaction process that was used for Climate Change, and the agreement statistics associated with it. Finally, in Sec C.3, we present detailed results for Climate Change, including simulated interactions. All results and conclusions are comparable with the Black Lives Matter results, showing that our approach generalizes across events.

C.1 Data Collection

We used the same Data Collection process for Climate Change as Black Lives Matter, discussed in Sec 5.1 and Sec A.1. This means we have the same

Model	E1-1 Acc	E1-1 F1	E1-2 Acc	E1-2 F1	E2-1 Acc	E2-1 F1	E2-2 Acc	E2-2 F1	# Edges
CLM No Interactions	40.16	32.77	39.65	31.86	34.88	30.93	-	-	-
CLM Sim. Interactions on 100% of Data in E1-1 + E2-1	46.72	41.59	39.65	31.86	44.18	41.57	-	-	12,602
CLM Human Interactions in E1-1 + E2-1	46.72	43.94	39.65	31.86	39.53	36.95	-	-	47
CLM No Interactions Train	-	-	49.29	44.84	44.77	42.35	-	-	-
CLM Sim. Interactions on 100% of Data in E1-1	-	-	56.33	50.02	44.18	43.10	-	-	12,602
CLM Human Interactions in E1-1	-	-	53.52	44.53	40.29	46.38	-	-	47
CLM No Interactions Train	-	-	49.29	44.84	44.77	42.35	44.44	33.06	-
CLM Sim. Interactions on 100% of Data in E1-1 + E1-2	-	-	56.33	50.02	41.86	36.42	34.04	24.93	12,602
CLM Human Interactions in E1-1 + E1-2	-	-	53.52	44.53	53.48	43.07	46.80	38.73	47

Table 9: Climate Change Results: Key: E1 and E2 are the two inductive graphs. E1-1/E2-1 is the first half, and E1-2/E2-2 are the second half. # Edges shows the number of edges added by interactions to E1-1. **Protocol 1:** The top third refers to Protocol 1, where interactions result in performance improvements in the difficult, inductive, no training setting. In particular, we see improvements of 6.56% Acc. and 11.17% F1 on $E1$-1. We do not evaluate on $E2$-2, as no interactions are done on that split, so performance does not change. **Protocol 2:** The middle third refers to Protocol 2, where interactions result in performance improvements when we train on the interactions, and then apply the model to a new event with no interactions done. We do not evaluate on $E1$-1, as it is the training set), and $E2$-2, as no interactions are done and performance does not change. **Protocol 3:** The last third refers to Protocol 3, where interactions result in performance improvements when we train on the interactions and then do more interactions in the inductive setting ($E2$-1). We also see improvements in $E2$-2. We don't evaluate on $E1$-1 as it is the training set.

Dataset	Low	Mixed	High
E1-1	30	43	49
E1-2	28	41	47
E2-1	11	15	17
E2-2	13	16	18

Table 10: Number of sources in our dataset for climate change.

3 time periods for Climate Change as Black Lives Matter (BLM). The only difference between BLM and Climate Change are the search terms we used to search Twitter to collect the data. For Climate Change, we used the following search terms: *fracking*, *global warming*, *climate change*, *#savetheplanet*, *#savethetrees*, *#climatechangeisreal*, *#waterpollution*, and *#climatestrike*. Statistics for the number of sources in each data split and their high, mixed, and low factuality distribution are shown in Table 10.

C.2 Human Interaction Process

For the human interactions done on Climate Change, we used two male human interactors. Both are Computer Science P.h.D. students in Natural Language Processing of Asian descent. The interactions were done on 8 sub-graphs for each time period ($E1$-1, $E2$-1, and the development set), for a total of 24 sub-graphs interacted on. As with Black Lives Matter, interactors spent an average of about 3 minutes on each interaction sub-graph.

C.3 Climate Change Results

In-depth results for climate change are presented in Table 9. In the main paper, due to lack of space, we presented only baseline and human interaction results, which we expand upon here also showing simulated results. We can see that results improve and are consistent with Black Lives Matter Results for Protocols 1, 2, and 3. Protocol 1 (top third of the table) shows significant Accuracy and F1 improvements in the fully inductive setting, showing the power of minimal human interactions in the right places to improve the model without any training. Protocol 2 (middle third of the table) shows how interactions result in performance improvements when we train on interactions and then apply them in the fully inductive setting with no additional interactions done. Finally, protocol 3 shows improvements when we train on the interactions and then do more interactions in the inductive setting.

The results in this section, combined with the earlier results on Black Lives Matter, show that our approach can generalize across multiple datasets, topics, and events.

Model	Embedding Change %
P1: Inductive Human	75.39
P2: Train Human	64.23
P3: Train + Inter. Human	51.41

Table 11: Change of node embeddings after interactions compared to the no interaction model on E2-1. Interactions affect model representations (lower # = more change).

D Discussion Continued

D.1 Model Representations

Now, in order to measure the impact of interactions on our graph-based model, we evaluate how much model node embeddings change after they are incorporated. To do this, we compute the difference in the cosine similarity of the embedding of each user node before and after interactions are done, and average the results. The results in Tab. 11 show that even a small amount of interactions can make a significant change in model representations. This shows why minimal amounts of interactions can lead to a strong performance increase.

D.2 Discussion: Interaction Examples

In this section, we continue our discussion from Sec 6.2 and provide more examples of nodes that humans connected during the interaction process. We first show the connections humans make (Sec. D.2.1), and then discuss what trends we can learn from these connections about our approach (Sec. D.2.2). The connections reveal interesting insights on how humans are connecting nodes based on content preferences. Further, it shows that despite all the content being focused on one event, there are lots of different relevant perspectives identified by the model as realistic points of confusion.

D.2.1 Interactions Made

We first show examples of pairs of users/articles that were connected by human interactors, describing what they were about. This analysis was done by the authors based on the human interactions.

1. A user with hashtags about taking back the United States by burning and destroying it, and also White Supremacy related hashtags, was connected to an article saying the current President (Biden) was clueless and didn't know what they were doing.

2. Two users with random and unrelated hashtags in their bio and extremely similar tweet language were connected as they were identified to likely both be bots.

3. A user that was a sports fan was connected to a source that reported sports media, but in this case had posted an article about how certain races have been negatively impacted from the coronavirus despite being athletic.

4. An article discussing how the Minnesota Vikings Honored George Floyd's family at their season opener was connected to a source that reported football sports articles that seemed factual.

5. An atheist, socialist, songwriter, and musician student Twitter user was connected to a Bernie Sanders supporter that wanted student loan forgiveness.

6. An influencer who was the mayor of a major city was connected to a seemingly politically aligned news reporter for the same city.

Next, we show snippets (to preserve anonymity) of user bios and articles that were connected, to show how simple the process is. We also provide our explanations of why these users/articles were connected. All of these examples are related to the Climate Change event and the text shown is snippets of the actual text that was shown to humans:

1. **Bio 1:** "what makes you optimistic...sharing optimism of optimistic leaders" **Tweet 1:** "a majority of young people are #optimistic that it's still possible to prevent the worst effects of #climatechange"

   Bio 2: "Christian...#Goodnews seeker, ther's plenty of it!"

   Explanation: These users were connected by interactors likely because the second user likes good news, and the first user is an optimist specifically sharing good news about climate change!

2. **Article 1:** "...San Diego May Get Climate Update After All.."

   Article 2: "Fish prices spike as ...face total depletion"

   Explanation: These articles were connected by interactors likely because they both are showing the effects of climate change. It is changing cities, and changing fish prices.

3. **Tweet 1:** "Climate Change...Biggest Hoax in Human History"

Tweet 2: "Trump is Hurting Climate Change by letting China take the lead..."

Explanation: These users were not connected (and so weren't the corresponding articles), and specifically marked **different**. This is likely because the first user doesn't believe in climate change, while the second is disappointed that President Trump isn't taking more action about it.

4. **Article 1:** "Climate Change...Biggest Hoax in Human History"

   Article 2: "California bans sale on new gas-powered cars in 2035"

   Explanation: These articles were not connected, and specifically marked **different**. This is likely because the first doesn't believe in climate change, while the second one does, or at least enough to report on the ban of the sale of gas cars to protect the environment.

D.2.2 Interactions Task Details and Trends

While humans can be subjective and make mistakes, we specifically designed our interaction task to be simple to try and eliminate as much of this as possible. Humans were asked to determine user similarity based on how users are discussing certain events, not in depth questions like if a text is factual or not. Determining this high level of user similarity is fairly simple, especially for educated humans, whom we envision performing the interactions.

From these examples above, we can see that our goal to reduce the subjectivity and increase simplicity of our interaction task holds true, at least in our experiments. This is why the entire interaction process can be be done rapidly (humans spent 3 mins per interaction graph, leading to the creation of 8 edges) and with high human interactor agreement. From the examples shown, it is clear that users/articles were connected based on content match, which was fairly simple for our educated human interactors to tell. However, this is hard for models, particularly on emerging news events, which is why our interaction setup leads to large performance improvements, even without any training. Also, we note that in most cases, the text defining the user/article similarity was not very subjective, and it is easy to determine the user/article perspective.

It is also possible, but unlikely, that two users/articles making similar statements don't have at least some similarity on an issue, and thus shouldn't be in the same information community. However, on a large scale over a lot of interactions, the text we show humans is likely to capture user/article perspectives and thus content similarity trends. Thus, even if there are a few rare cases in which users are connected but their statements aren't representative of the community they belong to, it isn't likely to make a significant difference in our learned representation and thus source factuality detection performance.

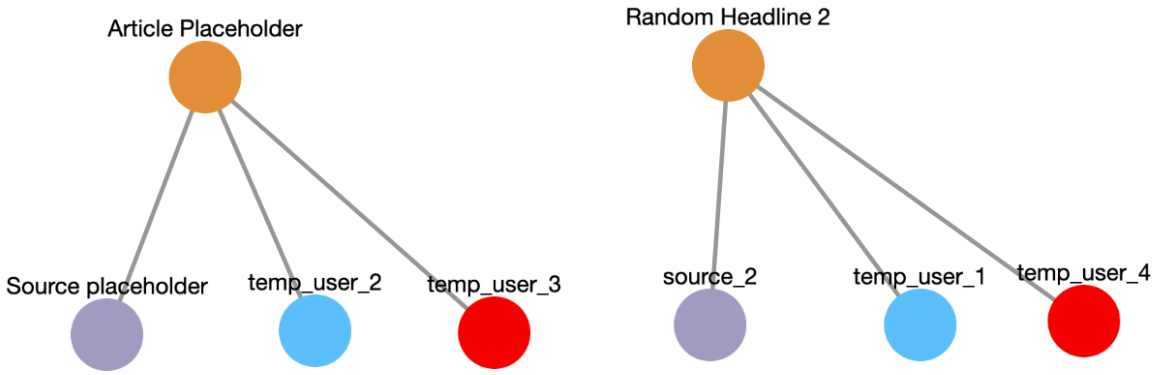

Figure 2: Example of an interaction graph that has been anonymized. The two red nodes are the pairs of users that were identified in Sec 4.1, shown by the Twitter usernames. The two orange nodes are the article nodes, and shown by their headlines without determiners. Blue nodes are other users that propagate the same articles (could be celebrities - users with over 1000 followers, and purple nodes are sources)

.

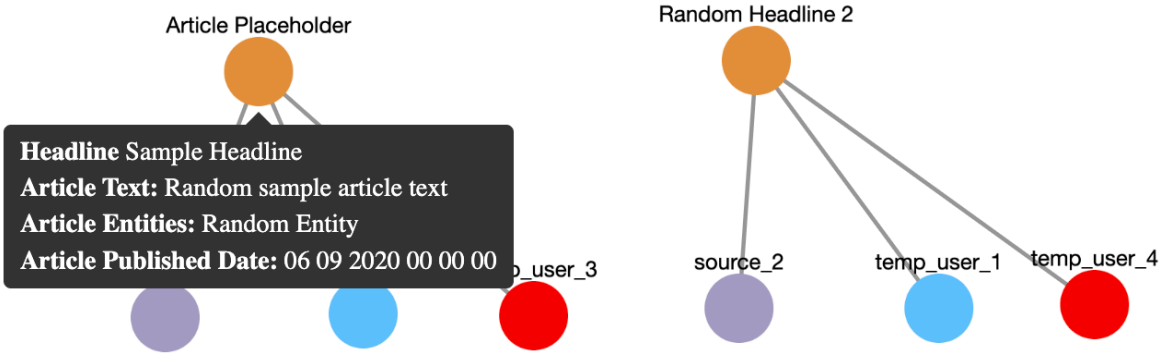

Figure 3: Example of an interaction graph where we can see metadata about an article, by clicking on the article node. This would be filled in during real human interactions, to allow humans to analyze the article and the context around it, but is currently anonymized.

.

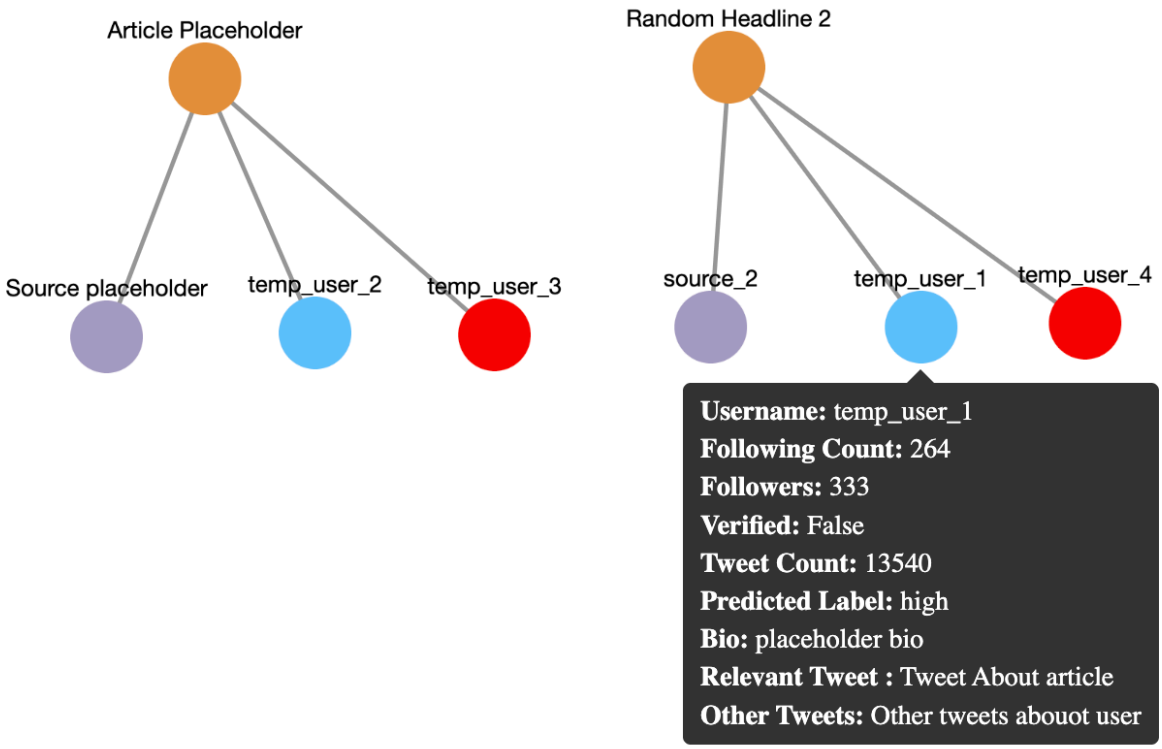

Figure 4: Example of an interaction graph where we can see metadata about a user, by clicking on the user node. This would be filled in (with data from Twitter) during real human interactions, to allow humans to analyze the user and the context around it, but is currently anonymized. Source nodes with Twitter profiles would appear with the same metadata.

.