Using RL to Identify Divisive Perspectives Improves LLMs Abilities to Identify Communities on Social Media

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

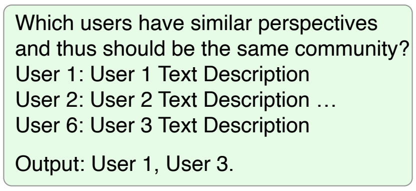
The large scale usage of social media, combined with its significant impact, has made it increasingly important to understand it. In particular, identifying user communities, can be helpful for many downstream tasks. However, particularly when models are trained on past data and tested on future, doing this is difficult.

In this paper, we hypothesize to take advantage of Large Language Models (LLMs), to better identify user communities. Due to the fact that many LLMs, such as ChatGPT, are fixed and must be treated as black-boxes, we propose an approach to better prompt them, by training a smaller LLM to do this. We devise strategies to train this smaller model, showing how it can improve the larger LLMs ability to detect communities. Experimental results show improvements on Reddit and Twitter data, on the tasks of community detection, bot detection, and news media profiling.

1 Introduction

The rise of social media platforms over the last decade has had a tremendous impact on people's lives, affecting their perspectives on key events such as political elections (Mitchell et al., 2016; Shu et al., 2019) and led to the creation of segregated information communities, also known as *"echo chambers"* (Gentzkow and Shapiro, 2011; Quattrociocchi et al., 2016; Dubois and Blank, 2018; Garimella et al., 2018). Following the the principal of *social homophily* (McPherson et al., 2001; Bessi et al., 2016), these tightly-knit communities consist of like-minded users, which have similar viewpoints and content preferences.

Identifying these information communities can lead to better performance in a number of important social media related downstream tasks, such as news media profiling (fake news and political bias detection), user content recommendation, trend prediction, crisis monitoring, sentiment analysis, and more (Bedi and Sharma, 2016). For example, for media profiling, groups of users sharing left-biased news in the past, are likely to do so in the future.



Which users have similar perspectives
and thus should be the same community?
User 1: User 1 Text Description
User 2: User 2 Text Description …
User 6: User 3 Text Description

Output: User 1, User 3.

Figure 1: An example of the LLM Community Detection Task: Given a set of users and their textual descriptions, determine which users are similar and have similar perspective.

The community identification task is typically formulated as a form of graph analysis, either predicting missing edges (i.e., friendship relationships), graph clustering (i.e., community detection), or more recently with deep learning, such as using graph neural networks (GNN) (Liu et al., 2020). However, due to the diversity of content found on social media, understanding users' perspectives using a fixed training set is highly challenging. For example, in the settings of **emerging news events**, the system is evaluated on its ability to adapt to new events, consisting of previously unseen users and topics. This temporal and topic shift at test time, hurts the performance of many models, and they must be retrained (Zhang et al., 2023). Since these settings are highly realistic (new topics and events emerge on social media everyday), we focus this paper on them and we evaluate these settings across a range of social media-related tasks.

In this paper, we explore a new direction for tackling such social inference tasks, inspired by the recently popular Large Language Models (LLMs), such as ChatGPT (OpenAI, 2022), which perform well on many NLP tasks. Specifically, given their ability to assess textual similarity well (OpenAI, 2023; Li and Li, 2023), we ask – ***can the strong textual similarity performance extend to the task of community detection?*** Given a set of users and text describing their viewpoints, we explore whether

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 5371–5390
November 12-16, 2024 ©2024 Association for Computational Linguistics

LLMs can identify if any of the users are similar. This way, social inference is reduced to a simpler text similarity problem (comparing user's text descriptions), and LLMs can help us form information communities. Fig. 1 shows an example of this community detection LLM task.

Intuitively, given their massive training datasets, LLMs have the potential to generalize across time periods and events, identify users with similar viewpoints, and thus perform well in the important emerging news events settings. However, we find that this task is still difficult for LLMs. We noticed that LLMs often focus on the high-level aspects of users to determine if they belong in the same community, favoring similarity of interest topics rather than nuanced opinions about them. As a result, LLMs often do not form meaningful communities. For example, two users discussing a popular entity like "Donald Trump", could be considered similar by a LLM, when in reality it's the context and attitudes expressed towards "Donald Trump" that makes them similar or not. If instead the LLM focused on how the users discuss Donald Trump (for example, their opinions on Trump's perspective on issues like gun control) then the LLM could correctly separate users into meaningful communities.

Our key technical contribution follows this intuition. We hypothesize that **focusing the LLM on the relevant aspects of users would result in better information communities**. We propose several models for automatically adding to the LLM prompt the exact topics and entities it should focus on to separate users into an information community. With the help of this additional information, the LLM can compare the user descriptions, focus on the divisive issues, and form the correct community. We call this additional prompt sentence a **focus area**. For example, in the running Donald Trump example, the focus area could be: *Focus on how the users discuss Donald Trump's views on gun control*. Tab. 8 show more ex. of good focus areas, and Fig. 2 shows how they can be useful.

Since many of the best performing LLMs are only accessible through an API, or are too large for task-specific training, we treat these models as black-boxes, and train a smaller LM to generate the focus area. This approach offers several advantages, such as being directly usable on top of any LLM, without changing the LLMs performance. We compare several variants of our approach, using the LLM directly (without focus areas), using the LLM to generate the focus areas and finally, training the smaller LM to generate the focus areas and augment the LLM prompts. We train the smaller LM using Reinforcement Learning (RL). The reward signal used by the RL algorithm is obtained by combining several rewards, such as the performance of the LLM when using the generated focus areas and "unsupervised" metrics capturing focus area topic relevance, informativeness, impact, length, and more (see Sec. 3.4.1 for details).

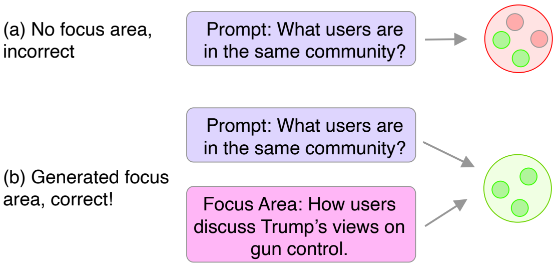

(a) No focus area, incorrect
Prompt: What users are in the same community?
(b) Generated focus area, correct!
Prompt: What users are in the same community?
Focus Area: How users discuss Trump's views on gun control.

Figure 2: An example of how Focus Areas can help. Without them (a), the LLM incorrectly forms the community (red users), but with them (b), the LLM focuses on the divisive issues and correctly forms the community (green).

We evaluate our approach in two settings. First, we define an intrinsic evaluation over Reddit and Twitter data, where users are sampled from known communities. Our goal is to recover the ground-truth community memberships via the focus-area augmented LLM prompts. Second, we look at the contribution of a focus-area augmented LLM based approach for downstream tasks that require social information – identifying false information and political bias in news media. Here, the gold community membership is unknown and can only be gauged by its contribution the downstream task. In both settings we model the out-of-domain emerging news event settings, by training the focus-area generator on a single community, and using it to generate focus areas for new, unseen communities.

In short, we make the following contributions: **(1)** We propose to use large, frozen LLMs to detect information communities on social media. **(2)** We train a smaller LM to generate a focus area, an additional prompt sentence to feed into the bigger LLM, to better detect information communities. To train the LM, we devise a novel Supervised and RL training procedure, specific to the social media setting. **(3)** We show how better community detection can improve the performance of downstream social media tasks in the challenging settings of emerging news events, specifically community detection, bot detection, and news source profiling (factuality/bias detection). We use Reddit and Twitter data.

Sec. 3 describes our framework, Sec. 4 our results, Sec. 5 analyzes, and Sec. 7 concludes.

2 Related Work

Over the last few years, there has been a number of works analyzing social media, whether it is news media profiling (Baly et al., 2018, 2020), fake news detection (Mehta et al., 2022; Yang et al., 2023), Reddit analysis (Arazzi et al., 2023), Bot Detection (Tan et al., 2023), or topic analysis (Roy and Goldwasser, 2023). These works utilize a variety of ML frameworks, such as LLMs (Su et al., 2023) and graphs (Phan et al., 2023; Ali et al., 2023), and evaluate a variety of settings such as cross-domain (Shu et al., 2022) and low-resource (Lin et al., 2022) ones. A more realistic and more challenging setting to analyze, which we also do, is one in which test samples mention different topics and feature different users than seen at training time. Due to their importance, these settings have also recently received more attention (Zhang et al., 2023; Mehta and Goldwasser, 2023a,b).

An important part of social media analysis is analyzing the users on social media. Specifically, prior work (Bessi et al., 2016; Ali et al., 2023) shows how grouping the users into information communities can provide insight for downstream tasks, such as fake news detection (Mehta et al., 2022), content recommendation (Singh et al., 2022), or even general analysis (Aguilar-Gallegos et al., 2022) such as how users view major events(Hao et al., 2024). In general, understanding user perspectives and forming these communities, is important, see: App. A.

Large Language Models (LLMs) have been applied to a large amount of social media related tasks, like fake news detection (Su et al., 2023), as they can capture a large amount of knowledge learned from their extensive pre-training. While they can succeed at many NLP tasks like summarization (Pu et al., 2023), they still struggle on reasoning tasks like needed for social media analysis. However, as we later show, when appropriately prompted, their performance on these tasks improves.

Prompting LLMs has been studied in a variety of ways, whether it be chain-of-thought reasoning (Wei et al., 2022), chain-of-hindsight (Liu et al., 2023), self-refinement (Madaan et al., 2023) or RLHF (Sun, 2023). Similar to Akyurek et al., we aim to train a smaller language model to prompt bigger, frozen LLMs. Similarly, improving LLMs using feedback has also received increasing research attention, across a variety of tasks, such as summarization (Ma et al., 2023; Yao et al., 2023; Hu et al., 2023). However, compared to tasks like summarization and action planning, social media analysis requires a more nuanced analysis, which affects the way we train our models (i.e. reward functions), and the feedback we provide.

3 Model

Our goal in this paper is to improve big, frozen LLMs performance on social media related tasks, specifically detecting user communities, as described in Sec. 3.1. To do this, we train a smaller LLM to add additional text, which we call a "focus area" (Sec 3.2), to the prompt of the bigger one. We train the smaller model first using Supervised Learning (Sec. 3.3), and then Reinforcement Learning (Sec. 3.4). Similar to Akyurek et al., we refer to the bigger, frozen LLM as LLM$_{\text{task}}$, and the smaller one as LLM$_{\text{prompt}}$.

3.1 User Community Detection

As mentioned in Sec. 1, detecting user communities has many advantages, such as understanding social media, content recommendation, etc. Moreover, using frozen LLMs to do this can bring further benefits, such as generalizing to new domains, avoiding fine-tuning big models, etc. Thus, in this section, we describe how we formulate the community detection task for frozen LLMs.

As the big, frozen, LLM$_{\text{task}}$ model can't be trained, it must be prompted. However, LLMs have limited context size, so we cannot prompt them with all the users on social media. Thus, we instead define the following, more simplified **community detection task**, which can be extended: Given a set of six users $U = u_1, ...u_6$, each with a textual description describing them, determine which, if any, users are similar to each other and should be in the same community $c_1 = u_1, ...u_c$.

LLM$_{\text{task}}$ responds in natural language, listing the users that are in the same community, and the ones that aren't. Fig. 1 shows a shortened example of this task, including our prompt to LLM$_{\text{task}}$, and App. B provides details (including generalization).

The textual description of each user in the prompt to LLM$_{\text{task}}$ is formed based on their social media posts, and provides information to LLM$_{\text{task}}$ to help it determine the user similarity. To form it, we prompt Chat-GPT to create a summary of the user given their posts (Twitter tweets, Reddit posts, etc.). We form this summary as it simplifies the community detection process, capturing the key details of the users viewpoints, while also being simpler to analyze than the individual posts. To ensure a relevant summary, we sample posts from users so that all six users U discuss at least one entity in common. An ex. of the LLM_{task} prompt we use is shown below in Tab. 1, Fig. 4, and App. B.

We note that in this setup, we ask LLM_{task} to detect a max of one community, placing all other users after, or not in a community. This setup can handle real-world settings, where there may be multiple, one, or no user communities in the users presented to the LLM. If there are multiple, LLM_{task} should form the most tightly-knit community.

Format	Language
Chat-GPT Question	What is the user discussing ... what is their perspective?
Input	Reddit Comment: ...
Output	The user mentions ...

Table 1: The question, text, and output format we use to create user summaries using LLM_{task} (shown for Reddit).

3.2 $\text{LLM}_{\text{prompt}}$ Definition: Focus Areas

In order to improve LLM_{task}'s ability to detect communities, we provide it an additional sentence as part of the prompt, which we call a **focus area**. The focus area tells the LLM exactly what to focus on when reading the user summaries, in order to properly separate the users into communities. We define this focus area to be a short sentence that details the divisive issues and topics that the current set of users are discussing. The focus area significantly simplifies LLM_{task}'s job, as it now just has to compare the user summaries based on the issues provided, to determine the community. Moreover, it makes sure LLM_{task} does not focus on high-level topics when determining user similarity, but rather on divisive issues. For ex., a focus area could be: *Focus on tax increase in California* (more: Tab. 8).

3.3 $\text{LLM}_{\text{prompt}}$: Supervised Training

To generate the focus areas, we train a smaller LLM, $\text{LLM}_{\text{prompt}}$, similarly to Akyurek et al.. We initialize it as an encoder-decoder model and fine-tune it to generate focus areas, given user summaries. We use T5-Base (Raffel et al., 2020), with 223M params, and then train on gold focus areas.

We approximate the gold focus areas using the gold communities and LLM_{task}, prompting it to generate the focus area based on the user summaries. Specifically, since we know the gold communities from the training data, we ask LLM_{task}: *What topics separate the gold communities?* Since LLM_{task} is told what the gold communities are, it is able to consider what separates the users to form the gold communities, and generate an initial focus area. We show an example in Tab. 2, a detailed example in Fig. 5, and provide details in App. F.

Format	Language
LLM_{task} Question	What topics should we focus on to determine first 3 users are in a community, while others are not?
Input	User_1 Summary, ... User_n Summary
Output	Focus on ...

Table 2: The question, input text, and output format we to create gold focus areas.

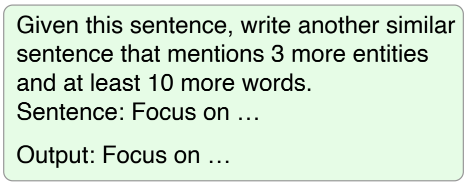
3.4 $\text{LLM}_{\text{prompt}}$: Reinforcement Learning

The supervised training phase above initializes the model to generate focus areas, but unfortunately, due to the gold data, many are still too high-level, and thus can be improved, for better community detection. Further, the gold data used to train $\text{LLM}_{\text{prompt}}$ comes from LLM_{task}, and our goal is to improve LLM_{task}'s performance. Thus, we must train $\text{LLM}_{\text{prompt}}$ directly on community detection, which we do using LLM_{task}'s predicted community outputs, when the output focus area from $\text{LLM}_{\text{prompt}}$ is used. However, as LLM_{task} is not trainable, we use Reinforcement Learning (RL), with several novel reward functions (RF), which we design specifically for community detection and describe in Sec 3.4.1. We then describe our curriculum learning RL training procedure in Sec. 3.4.2.

3.4.1 Reward Functions

We use 4 novel reward functions to train $\text{LLM}_{\text{prompt}}$ to generate better focus areas. To optimize them, we use the same training dataset as Sec. 3.3.

RF1: Coverage, Community Detection Performance: Our first reward, **Coverage**, described in detail in Sec. 4.3.1, optimizes community detection directly, thus learning focus areas that help improve community detection performance. Specifically, given two gold communities c_1, c_2, and two predicted communities p_1, p_2, the reward is: *How many users from each predicted community are part of the same gold community?* To compute this reward while ignoring the order of predicted communities, we first find the largest overlapping gold community for each predicted one, and then compute the overlap accuracy score for each. We note that while LLM_{task} is prompted to predict one



Given this sentence, write another similar sentence that mentions 3 more entities and at least 10 more words.
Sentence: Focus on …

Output: Focus on …

Figure 3: How we prompt ChatGPT to generate more informative focus areas (positive class), given ones from the training set (negative class). We then train a binary LR model on this data.

community (for simplicity), it still places the rest of the input users together in another community, and we have gold data for two communities, which is why this reward function evaluates both.

RF2: Entity Frequency: Our second reward, entity frequency, improves focus areas by getting them to mention entities that may be useful to separate users into communities. To do this, we find entities that are more frequently mentioned by one gold community compared to the others, and provide a reward based on how many of those entities the focus area mentions. Specifically, we first extract entities (Spacy NER-tagger (Honnibal et al., 2020)) from each user summary, keeping ones that are mentioned more than once across a gold community. Then, we find the entities that are mentioned more often by one of the gold communities. We provide a reward based on how many of these entities are mentioned in the generated focus area scaled to a max of 3 (i.e. 3+ entities = 1.0 reward).

RF3: Focus Area Informativeness: This reward function scores focus areas, aiming to make them more informative, so they capture more details about communities. This is essential, as our motivation for providing focus areas to LLM$_{\text{task}}$ is to make it not rely on general topics, but rather details, to determine communities. To score focus areas, we train a Logistic Regression model on data generated using ChatGPT. We use gold focus areas as negative examples, and for positive examples, we prompt ChatGPT to generate more informative versions of the gold focus areas (as seen in Fig. 3).

RF4: Focus Area Length: Our final reward function optimizes focus areas to be longer in length, so they can capture more details. We determine the number of words in the predicted focus area, provide 0.5 reward if it is less than 10, 1.0 if it's more than 35, and otherwise a value that scales linearly between 0.5 and 1.0 (up to 35 words).

3.4.2 Curriculum Learning

We finetune LLM$_{\text{prompt}}$ using Proximal Policy Optimization (Schulman et al., 2017) and the reward functions above, using the implementation by (Akyurek et al., 2023; Ramamurthy et al., 2022).To stabilize the learning of the reward functions from above (Sec. 3.4.1), we use curriculum learning. Alg. 1 provides pseudo-code for our overall training process, App. D details of RL + Reward Functions, and App. E details of curriculum learning. Our rewards balance each other, i.e. generating useful, entity relevant, informative, and longer focus areas.

Dataset	Train	Val	Test
Reddit Politics	2,789	100	550
Reddit Economic	-	-	232
BotPercent	-	-	155
Twitter	-	-	444

Table 3: Dataset size statistics. Each sample has 6 users, and all test users are unique across samples.

Algorithm 1 *Algorithm to Train LLM$_{prompt}$ to Generate Focus Areas*

```
Input: LLM$_{\text{prompt}}$, LLM$_{\text{task}}$ (Initialized Prompt Model, Frozen Task Model)
Input: Dataset $\sum_{i=1}^{n} D = (u_1...u_6, c_1, c_2, f)$ (Users $u_1, ...u_6$ to separate into communities $c_1, c_2$ and Gold Focus Area $f$ to train LLM$_{\text{prompt}}$
Output: LLM$_{\text{prompt}}$ (Trained Focus Area Generation Model)
Supervised Training: Maximize $f$ : $\mathbb{E}\left[\log p_\theta(f|u_1, \ldots, u_n)\right]$ (Train LLM$_{\text{prompt}}$ to generate focus areas)
while not converged do
    Sample mini-batch: $\sum_{i=1}^{n} D = (u_1...u_6, c_1, c_2)$
    Generate focus area: $\hat{f} \sim$ LLM$_{\text{prompt}}(u_1...u_6)$
    Use Focus area to get community prediction: $\hat{c_1} \sim$ LLM$_{\text{task}}(u_1...u_6, \hat{f})$
    Get Reward Based on Community Prediction: $R =$ Reward$(c_1)$
    Update LLM$_{\text{prompt}}$ based on reward $R$
end while
return LLM$_{\text{prompt}}$ (Trained Focus Area Generation Model)
```

4 Experiments

4.1 Datasets

Our goal in this paper is to improve big, frozen LLMs (LLM$_{\text{task}}$) ability to detect communities. Specifically, given a set of six users with their profile/post summaries, LLM$_{\text{task}}$ should be able to detect which (if any) users belong to the same community. We now describe our evaluation datasets, including on downstream tasks (4.1.3). Tab. 3 shows the number of samples in our different datasets. We release our data [1].

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

4.1.1 Reddit

Our first dataset, collected by us, directly evaluates how well LLM_{task} can detect communities. To get the gold data, we use the social media site Reddit.

Reddit is made up of communities called subreddits, each of which consists of posts relating to a central topic, such as "Politics". Reddit users make these posts, and other users interact with the posts by commenting or voting on them (up-vote or down-vote). Each subreddit additionally has designated moderators, users who monitor the subreddit, performing actions such as deleting posts that are not relevant to the subreddit. Further, users often down-vote posts that disagree with the ideas of the subreddit. Thus, subreddits and their up-voted content are very similar to real life communities, as they contain similar minded users that discuss topics relevant to the central theme of the subreddit.

Building on this, we hypothesize that users in the same subreddit, who have a positive up-vote score across all their posts in the subreddit, are members of the subreddit's community. Thus, a set of users from one subreddit form one community, and a set of users from another from a different community, and LLMs should be able to tell the difference.

We build two datasets to evaluate this, sampling data from two polarizing subreddits, or communities. The first (Political) dataset is from the "Democrats" subreddit and the "Conservative" subreddit, while the second (Economic) is from "Capitalism" and "Socialism". Each dataset sample has six users across two communities (three from the first subreddit/community, and three from the second), which must be separated. To construct each sample, we find two posts, one from each subreddit, that discuss the same topics (made within three weeks of each other and their titles' having at least one entity in common (Akbik et al., 2019)). For each post, we sample three users that belong to the subreddit and comment on the post. As long as their comments have a positive up-vote score, we know that these three users and post is representative of that subreddit's community. After doing this for both subreddits, we obtain a total of six users, three from one subreddit community and three from another, which forms a sample for our dataset. After creating summaries for each user based on their post comments (as discussed in Sec. 3.1), we can ask the LLM to detect the communities.

4.1.2 TwiBot

Our second dataset also evaluates how well LLM_{task} can detect which users in a given set of six users are in the same community. However, this dataset is from Feng et al., and evaluates whether Twitter users are bots or not. The dataset, named TwiBot-20, consists of Twitter users, their metadata (tweets, profile information), and a label signifying whether they are bots or not. The dataset additionally groups users into four broad categories: Politics, Business, Entertainment, and Sports. We construct test samples, each with six users from two communities, using this dataset, where each sample has users belonging to the same category, and the two communities are bot and not bot. While other works (Feng et al., 2022; Tan et al., 2023) also used this dataset, we do not compare to them directly, as our setup is unique to our task (other works use graphs, etc. which we evaluate in Sec. 4.1.3).

4.1.3 News Source Profiling

Our final evaluation is on downstream tasks, showing how detecting communities can improve news source profiling (factuality/bias detection). We use the dataset originally proposed by Baly et al. (2020, 2018) and also evaluated by Mehta et al. (2022).

The dataset consists of sources scraped from Media Bias/Fact Check[2], each labeled on a 3-point scale for factuality (high, low, mixed) and bias (left, center, right). Following prior work (Baly et al., 2020), we aim to predict the factuality/bias of the news sources using Twitter data, which provides social context. It consists of sources (the classification targets), the articles they publish, and users who interact with the sources or articles (propagate the articles, follow users/sources). Following Mehta et al., we build an information graph using this data. We follow the challenging fully inductive evaluation protocol proposed by Mehta and Goldwasser (2023a), where the test set graph is not connected to the training set graph in any way (no users, sources, articles or edges in common).

Similar to Mehta et al., we hypothesize that detecting user communities can increase profiling performance. This is because, similar users are likely to have similar views and thus spread similar content, which has similar factuality/bias. This has also been shown in social homophily theory (Bessi et al., 2016). Thus, we randomly sample groups of users, ask LLM_{task} to form communities., and con-

[2] https://mediabiasfactcheck.com

Dataset: Model	Coverage	# Test Samples
Reddit Political: No Focus Areas	42.01	550
Reddit Political: Gold (ChatGPT) Focus Areas	44.66	550
Reddit Political: $\text{LLM}_{\text{prompt}}$ Focus Areas: Supervised Learning	45.48	550
Reddit Political: $\text{LLM}_{\text{prompt}}$ Focus Areas: RL Curriculum Learning	**47.85**	550
Reddit Economic: No Focus Areas	42.25	232
Reddit Economic: Gold (ChatGPT) Focus Areas	44.60	232
Reddit Economic: $\text{LLM}_{\text{prompt}}$ Focus Areas: Supervised Learning	44.00	232
Reddit Economic: $\text{LLM}_{\text{prompt}}$ Focus Areas: RL Curriculum Learning	**45.58**	232
TwiBot: No Focus Areas	21.63	155
TwiBot: Gold (ChatGPT) Focus Areas	19.19	155
TwiBot: $\text{LLM}_{\text{prompt}}$ Focus Areas: Supervised Learning	22.55	155
TwiBot: $\text{LLM}_{\text{prompt}}$ Focus Areas: RL Curriculum Learning	**22.72**	155

Table 4: Results on Reddit Political, Reddit Economic, and TwiBot (Bot detection (Feng et al., 2021)) community detection datasets when using ChatGPT for LLM_{task} and T5-Base for $\text{LLM}_{\text{prompt}}$. All of this test data is in the unseen emerging news events settings, and features new topics published after the time period the training set was collected from. Using focus areas improves performance on all three datasets, and training $\text{LLM}_{\text{prompt}}$ using RL leads to the best performance on each dataset. This shows the benefit of our framework to learn useful focus areas, and those focus areas to improve community detection performance, even on domains and time periods not seen at training time.

nect users in the same communities in the graph.

4.2 Training/Test Procedure

We train **only** on our first Reddit dataset, which consists of politics subreddits: 'Democratic' and 'Conservative', collected between the start of 2013 and end of 2016. Thus, **we don't train/finetune on any of the other test datasets**. We provide details in App. E.1, and release our code and data.[3]

As discussed in Sec. 1, all of our test data is in the challenging **emerging news events setting**, which consists of topics and time periods not seen at training time. We first test on the two Reddit datasets, which feature posts made between 2018 and the end of 2023, and then TwiBot-20 (Feng et al., 2021). Finally, we evaluate news media profiling, which features posts from after 2019. Importantly, this evaluation is also in the fully inductive setting, so the test set graph does not have any users or nodes in common/connected to the training graph.

4.3 Evaluation Metrics

4.3.1 LLM_{task} Evaluation

To evaluate LLM_{task}'s ability to detect information communities, we use a comprehensive metric, which we refer to as **Coverage**. To compute it, we first determine the appropriate gold community. This is important, as LLM_{task} is only asked to predict one community, but the gold data has two. To evaluate, we choose the gold community as the one that has the largest number of overlapping users with LLM_{task}'s predicted community. We then determine how many users were correctly predicted, out of all the users both predicted and missing. Mathematically:

$$\frac{\text{\# of correct pred.}}{\text{\# of correct + incorrect + missing pred.}} \quad (1)$$

This metric prioritizes both predicting the communities correctly, and not missing any users.

4.3.2 News Source Profiling

For source profiling, we evaluate Accuracy and Macro F1 (the dataset is unbalanced) for news sources, using the dataset proposed by (Baly et al., 2020) and expanded by (Mehta and Goldwasser, 2023a) for the inductive test set settings.

4.4 Results

4.4.1 LLM_{task} Evaluation

Tab. 4 shows our results when we use ChatGPT as LLM_{task} on the two Reddit datasets (Political and Economic) and TwiBot Bot Detection (Feng et al., 2021). Tab. 7 shows results when Llama 2 is used as LLM_{task}, showing our framework generalizes across LLMs. We evaluate emerging news events, where test data is unseen and collected from time periods after the training data. On each dataset, focus areas lead to significant performance improvements, particularly our $\text{LLM}_{\text{prompt}}$ model after it is trained with RL and Curriculum Learning.

[3] Code and data released with this submission.

Model	FN Acc.	FN F1	Bias Acc.	Bias F1
(Mehta et al., 2022)	44.66	28.50	47.74	34.69
(Mehta et al., 2022) + LLM_{task} + Focus Areas	**45.53**	**30.17**	**48.64**	**36.34**

Table 5: News Source Media Profiling: Fake News (FN) and Political Bias Detection. When added via edges to the graph (1444 edges), the communities formed using ChatGPT for LLM_{task} + focus areas lead to improvements, showing the usefulness of LLMs to form communities which help downstream tasks, despite no training on this domain.

When evaluated on the same (but future) domain as training, Reddit Political, $\text{LLM}_{\text{prompt}}$'s focus areas lead to a 5.84% performance improvement in Coverage, with RL providing ∼5% relative improvement. On a different domain, economic data, performance improves 3.33%, showing the benefit of our framework to transfer to different domains. On Bot Detection, focus areas lead to more than 5% improvement. Thus, focus areas improve LLM_{task} community detection, even on unseen domains.

4.4.2 News Source Factuality Detection

Tab. 5 shows results on news source factuality detection. We evaluate 444 sources for factuality (183 high, 131 mixed, 128 low) and bias (202 right, 109 left, 108 center, rest unknown), and 212 comms. We compare to Mehta et al., but in the emerging news events settings, using the public Black Lives Matter data from Mehta and Goldwasser. We see that using LLM_{task} with Focus Areas to form communities leads to improvements (over 4% relative increase on Bias F1). This shows the benefit of using LLMs to form communities to improve downstream social media tasks, particularly when LLMs are prompted with focus areas. Details: App. C.

5 Discussion

We analyze our best RL $\text{LLM}_{\text{prompt}}$ model, with ChatGPT. We do an ablation study of our reward functions, (5.1), then a human analysis of generated focus areas, (5.2), then case studies (App. H), then an analysis of LLM detected user communities for factuality detection (App. I), and finally discuss the real world impact of our approach (5.3).

5.1 Ablation Study

Reward Fn.	Coverage	# Samples
None: No Focus Area	42.01	550
Coverage	46.07	550
Entity Frequency	46.96	550
Informativeness	46.90	550
Length	45.58	550
All: Curriculum Learning	**47.85**	550

Table 6: ChatGPT + T5-Base Reward Function Ablation Study on Reddit Political Data. Although each reward function leads to improvements, using all of them via Curriculum Learning performs the best.

App. J shows the benefit of RL, and Tab. 6 the results of our reward function ablation study. While we notice improvements compared to not using focus areas, they are not as significant, showing the benefit of RL and learning the rewards together. Doing so enables each reward function to contribute to learning an overall useful focus area.

5.2 Human Analysis of Focus Areas

We have 3 humans analyze 50 of $\text{LLM}_{\text{prompt}}$'s focus areas, comparing them to the ChatGPT generated ones. They score each focus area on a scale of 1-5, for grammatical correctness and usefulness (to identify divisive issues and user communities). On average, on grammar, ChatGPT scores 4.95, and $\text{LLM}_{\text{prompt}}$ 3.00. However, on usefulness, ChatGPT scores 3.07 and $\text{LLM}_{\text{prompt}}$ 3.26. From this, we see $\text{LLM}_{\text{prompt}}$ generates better focus areas to separate users into communities, which explains our results from Sec. 4. App. G provides details.

5.3 Real World Impact

Our framework to generate focus areas can be utilized with any LLM_{task} in the real world, even without fine-tuning it. This is because, focus areas are just an additional input to the prompt of LLM_{task}. Moreover, as we evaluated extensively on emerging news events, particularly on topics and tasks on which our models were not trained on (Reddit Economic, TwiBot, and Source Factuality Detection) our framework is very applicable in the real world on social media, where new topics arise daily. Most importantly, $\text{LLM}_{\text{prompt}}$ doesn't have to be retrained every time a new topic arises.

6 Acknowledgements

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

7 Conclusion

In this paper, we proposed to use large, frozen LLMs to detect user information communities on social media, particularly in the challenging settings of emerging news events, where test data features topics and time periods not seen at training time. We then improved this LLMs performance, by training a smaller LM ($\text{LLM}_{\text{prompt}}$) to generate a focus area, an additional sentence to feed into the bigger LLM. This focus area focuses the LLM on the relevant aspects of users that would result in better information communities, such as divisive issues. Experimental results on Reddit and Twitter data showed performance improvements in detecting communities when using Focus Areas, even on emerging news events. We evaluated four different datasets, all from time periods occurring after the training set, and three different domains, making the task very challenging. Further, we learned meaningful communities, that lead to improvements on the downstream task of source profiling (factuality/bias detection).

Our future work is primarily to generate better focus areas. We are particularly interested in exploring different reward functions, such as using different off the shelf models, to improve the quality of focus areas and make them more useful. We also want to explore the impact of multiple stages of SFT and RL training, where focus areas can improve across stages. Finally, we aim to conduct our experiments at a larger scale, on bigger datasets, so we can further study the real world impact of our framework.

8 Ethics Statement

8.1 Limitations

In this paper, we proposed a framework to train and evaluate on social media data, specifically Reddit and Twitter data and English. The framework we presented, and the experimental results we achieved, are shown for these domains/tasks. We believe that they will generalize to other domains and tasks, but we leave the exploration of that to future work.

In this paper, we focused on the emerging news events settings, where we evaluated when the test data was not seen at test time. These are some of the most challenging settings for social media tasks, as knowledge learned at training time can't always be used at test time. This is also why we leveraged LLMs for this task. Our future work involves testing how our experiments in this paper can generalize to other domains of emerging news events.

In this paper, we used two Large Language Models: ChatGPT and Llama 2. For ChatGPT, we used the API released publicly by OpenAI, and the details of the model are not known. For Llama 2, we ran it locally, using the Llama-cpp-python library. We specifically run the 70B parameter model, as detailed in Appendix E.1. While we use both of these models as black-boxes, and they perform well in numerous benchmarks (Qin et al., 2023), we understand that our frameworks build on these models and this could be a potential limitation. We believe it's important to take caution when deploying these models.

For experimental reasons, we set up our framework to detect communities in sets of 6 users. We hypothesize this can generalize, to number of users more than or less than 6. Specifically, it there are less than 6 users, generalizing is simple, just provide less users in the prompt. If there are more than 6, our framework can be used by either breaking the number of users into groups of 6, and then asking the LLM to detect communities, or by just passing in more than 6 users at once. While we did not test the latter, we hypothesize it may still work provided the LLM has the ability to handle the longer context, and leave it for future work.

8.2 Ethics

We do not believe we violated any code of ethics in our experiments done in this paper. We release our full code and anonymized data, to make the reimplementation of our models as simple as possible. We also caution that our models are the output of a machine learning model, and this could be parameter/machine dependent.

In our Reddit dataset release, we anonymized all the user data, to violate no code of ethics. Further, the data we scraped was released publicly by (Chang et al., 2020). Thus, all the data we used is previously publicly available.

Our framework in general is to be used to analyze social media and form information communities along with LLMs. Our general experimental settings of forming focus areas may also be useful for other tasks, and we leave the investigation of this to future work.

Our framework also has the potential to be used in malicious ways, along with positive ones. Specifically, identifying users that belong to specific communities can potentially impact those users, even in harmful ways, such as if this knowledge is made public. While there are clear positives to our community detection approach, such as downstream tasks or finding ‘friends’ for other users, this is one of the downsides. Thus, our framework must be used with caution.

When considering our work, it’s important to consider these and other related things to make sure the usage of our framework and code/data release falls within appropriate and safe use.

References

Norman Aguilar-Gallegos, Laurens Klerkx, Leticia Elizabeth Romero-García, Enrique Genaro Martínez-González, and Jorge Aguilar-Ávila. 2022. Social network analysis of spreading and exchanging information on twitter: the case of an agricultural research and education centre in mexico. *The Journal of Agricultural Education and Extension*, 28(1):115–136.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.

Mohsan Ali, Mehdi Hassan, Kashif Kifayat, Jin Young Kim, Saqib Hakak, and Muhammad Khurram Khan. 2023. Social media content classification and community detection using deep learning and graph analytics. *Technological Forecasting and Social Change*, 188:122252.

Marco Arazzi, Serena Nicolazzo, Antonino Nocera, and Manuel Zippo. 2023. The importance of the language for the evolution of online communities: An analysis based on twitter and reddit. *Expert Systems with Applications*, 222:119847.

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.

Punam Bedi and Chhavi Sharma. 2016. Community detection in social networks. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 6(3):115–135.

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.

Jonathan P Chang, Caleb Chiam, Liye Fu, Andrew Z Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. Convokit: A toolkit for the analysis of conversations. *arXiv preprint arXiv:2005.04246*.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. 2023. Reinforcement learning can be more efficient with multiple rewards. In *International Conference on Machine Learning*, pages 6948–6967. PMLR.

Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *Proceedings of the national academy of Sciences*, 113(3):554–559.

Elizabeth Dubois and Grant Blank. 2018. The echo chamber is overstated: the moderating effect of political interest and diverse media. *Information, Communication & Society*, 21(5):729–745.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, et al. 2022. Twibot-22: Towards graph-based twitter bot detection. *Advances in Neural Information Processing Systems*, 35:35254–35269.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4485–4494.

Alex Fenton, Leah Gillooly, and Cristina Mihaela Vasilica. 2023. Female fans and social media: Micro-communities and the formation of social capital. *European Sport Management Quarterly*, 23(2):370–390.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 World Wide Web Conference*, pages 913–922. International World Wide Web Conferences Steering Committee.

Matthew Gentzkow and Jesse M Shapiro. 2011. Ideological segregation online and offline. *The Quarterly Journal of Economics*, 126(4):1799–1839.

Fei Hao, Eunhye Park, and Kaye Chon. 2024. Social media and disaster risk reduction and management: How have reddit travel communities experienced the covid-19 pandemic? *Journal of Hospitality & Tourism Research*, 48(1):58–83.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Jian Hu, Li Tao, June Yang, and Chandler Zhou. 2023. Aligning language models with offline reinforcement learning from human feedback. *arXiv preprint arXiv:2308.12050*.

Xianming Li and Jing Li. 2023. Deelm: Dependency-enhanced large language model for sentence embeddings. *arXiv preprint arXiv:2311.05296*.

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.

Fanzhen Liu, Shan Xue, Jia Wu, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Jian Yang, and Philip S Yu. 2020. Deep learning for community detection: progress, challenges and opportunities. *arXiv preprint arXiv:2005.08225*.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Chain of hindsight aligns language models with feedback. *arXiv preprint arXiv:2302.02676*, 3.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Eureka: Human-level reward design via coding large language models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

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 and Dan Goldwasser. 2023a. An interactive framework for profiling news media sources. *arXiv preprint arXiv:2309.07384*.

Nikhil Mehta and Dan Goldwasser. 2023b. Interactively learning social media representations improves news source factuality detection. *arXiv preprint arXiv:2309.14966*.

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.

Amy Mitchell, Jeffrey Gottfried, Michael Barthel, and Elisa Shearer. 2016. The modern news consumer: News attitudes and practices in the digital era. *Pew Research Center*.

OpenAI. 2022. GPT-3.5 (ChatGPT). Computer software.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. 2023. Fake news detection: A survey of graph neural network methods. *Applied Soft Computing*, page 110235.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The Eleventh International Conference on Learning Representations*.

Shamik Roy and Dan Goldwasser. 2023. “a tale of two movements’: Identifying and comparing perspectives in# blacklivesmatter and# bluelivesmatter movements-related tweets using weakly supervised graph-based structured prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10437–10467.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Kai Shu, Ahmadreza Mosallanezhad, and Huan Liu. 2022. Cross-domain fake news detection on social media: A context-aware adversarial approach. In *Frontiers in Fake Media Generation and Detection*, pages 215–232. Springer.

Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320.

Dharmendra Kumar Singh Singh, N Nithya, L Rahunathan, Preyal Sanghavi, Ravirajsinh Sajubha Vaghela, Poongodi Manoharan, Mounir Hamdi, and Godwin Brown Tunze. 2022. Social network analysis for precise friend suggestion for twitter by associating multiple networks using ml. *International Journal of Information Technology and Web Engineering (IJITWE)*, 17(1):1–11.

Jinyan Su, Claire Cardie, and Preslav Nakov. 2023. Adapting fake news detection to the era of large language models. *arXiv preprint arXiv:2311.04917*.

Hao Sun. 2023. Reinforcement learning in the era of llms: What is essential? what is needed? an rl perspective on rlhf, prompting, and beyond. *arXiv preprint arXiv:2310.06147*.

Zhaoxuan Tan, Shangbin Feng, Melanie Sclar, Herun Wan, Minnan Luo, Yejin Choi, and Yulia Tsvetkov. 2023. Botpercent: Estimating bot populations in twitter communities. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Sin-Han Yang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Entity-aware dual co-attention network for fake news detection. *arXiv preprint arXiv:2302.03475*.

Zonghai Yao, Benjamin Schloss, and Sai Selvaraj. 2023. Improving summarization with human edits. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2604–2620.

Yuji Zhang, Jing Li, and Wenjie Li. 2023. Vibe: Topic-driven temporal adaptation for twitter classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3340–3354.

A Importance of Community Detection

In this paper, we aimed to improve the performance of LLMs to detect communities of users on social media. Community detection is an important task in social media analysis, for several reasons. For example, if we know from historical data that a group of users have similar perspectives and are thus in the same community, then it's more likely that content shared by some of the users in the group will also be agreed upon by others in the group. This can be beneficial for downstream tasks. For example, for fake news detection, news shared by a community that historically shares fake news is more likely to be fake news. Further, users in that community are also more likely to share fake news. Thus, if we can identify the fake news sharing community, we have more knowledge about the users in that community, and we can identify new fake news content better. Similar ideas apply to political bias detection. This was also shown by (Bessi et al., 2016; Del Vicario et al., 2016; Mehta et al., 2022; Mehta and Goldwasser, 2023a).

Prior work has also shown how detecting communities on social media can improve other downstream tasks, beyond fake news/political bias detection. It can help us analyze trends on social media (Singh et al., 2022), such as how people view major events like COVID-19 over time (Hao et al., 2024). It can help us understand how different groups of people are treated, such as female sports fans (Fenton et al., 2023). It can also help us analyze hate speech on social media (Ali et al., 2023), which is important to maintain a healthy society.

For these reasons and more, the community detection task is very important, which is why we focused on it in this paper. However, as we showed in Sec. 4, community detection in the out-of-domain settings where test data is never seen before is challenging. As LLMs capture a large amount of external knowledge, and can thus generalize, these out-of-domain settings are where we can take advantage of LLMs to perform better, assuming we use them correctly. This is where focus areas significantly help, as they tell the LLM what topics/entities to focus on, in order to correctly identify communities, unlocking LLMs for this community detection task.

B Community Detection Task Details

As defined in Sec. 3.1, our community detection task is: Given a set of six users $U = u_1, ...u_6$, each with a textual description describing them, determine which, if any, users are similar to each other and should be in the same community $c_1 = u_1, ...u_c$.

We represent each user in the prompt to the LLM (LLM$_{\text{task}}$) by a textual summary, created using Chat-GPT, based on a textual description coming from the user's social media. To get this textual description, for Twitter users, we use 10 of their randomly selected tweets and their profile metadata (profile bio, number of followers, number of people following, number of likes, number of tweets, and whether they are verified or not). For Reddit users, we use the comments they made in relation to the post on the topic LLM$_{\text{task}}$ is analyzing to determine the community. We provide this textual data to Chat-GPT and ask it to summarize it. The exact summarization task, with prompt input, for Twitter users, is seen in Fig. 4.

We note that our task setup can easily generalize to any number of users bigger than 6, by breaking them up into groups of 6 and then asking the LLM. More importantly, this task setting is just a way that we set up the input to the LLM. We hypothesize that our entire framework will work with more/less users, and focus areas will be equally effective. Of course, our framework is reliant on LLM$_{\text{task}}$. Thus, if too many users are used and the LLM cannot handle the large context length, then the baseline and baseline + focus area community detection performance would suffer. We found that 6 was a good number of users for the LLMs we tested.

We also note that it's possible that there are no user communities in the groups of users presented to LLM$_{\text{task}}$. In this case, LLM$_{\text{task}}$ shouldn't be forced to detect a community. This is why we asked LLM$_{\text{task}}$ to form only one community, and place all the other users after (i.e. they don't belong to a community). For example, if there is no community, the LLM won't predict one, and just place all users after (the LLM uses a separator ';;;;;' to separate communities, so in this case the output would be something like: "$; ; ; ; ; user_1, user_2, user_3, etc.$"). On the contrary, if there are multiple communities, LLM$_{\text{task}}$ should form the single most closely-knit community. This is also a design decision, which future work can change.

Additionally, we note that the community detection process can be done in several steps, i.e. forming a community and narrowing it down in future iterations to be more topically focused, and leave the exploration of this for future work.

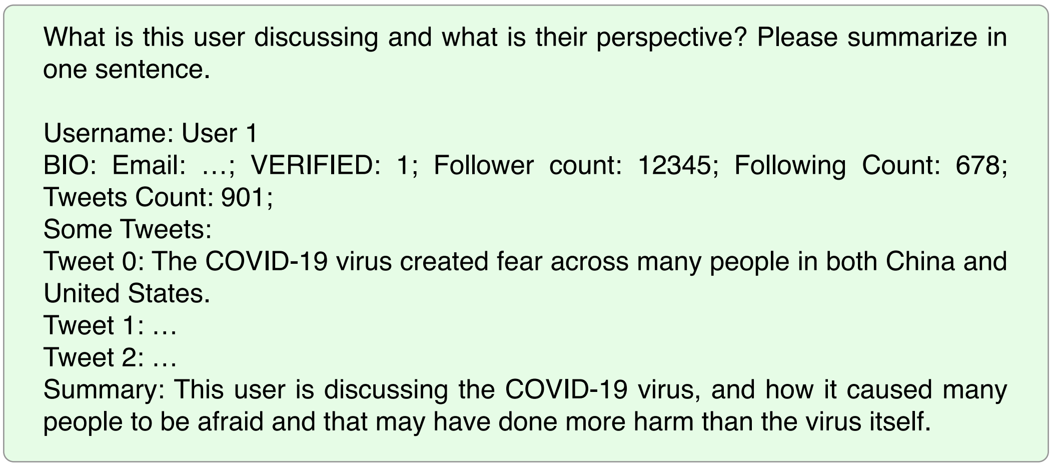
C News Media Profiling Details

For news media profiling, where we evaluate news source factuality and bias detection, we use the public information graph model from (Mehta et al., 2022), which they originally trained for only news source factuality detection. As we also evaluate political bias source detection, we train the model on both classification objectives. The model uses a Relational GCN to encode a graph structure, training it for the Source Node Classification objective. The Graph consists of three node types: Users, Sources, and News Articles, each with an initial representation (twitter information for users and sources, SBERT embedding article text for articles), which is updated throughout the training process. In the graph, articles are connected to the sources they come from with edges, while users are connected to sources they follow, or other users they follow. Thus, the users provide the social information in the graph, which we aim to better learn, by building better information communities.

Once the graph is trained, we evaluate it in the challenging fully inductive settings, where no test nodes are common with the training set, and no test nodes are connected to training set nodes. We aim to determine if strengthening the user relationships in the graph, i.e. building stronger information communities, can improve the performance on this challenging downstream task. To do this, we sample sets of six users that are close to each other in the learned Graph embedding space (this increases the chance that they discuss similar topics), and run them in our community detection approach. Specifically, we ask LLM$_{\text{task}}$ to either place some (or none) of these users into a community. For the users LLM$_{\text{task}}$ thinks are similar, we directly connect them in the information graph, using a user-user edge. We then evaluate our downstream tasks, again.

To build the set of six users, we first randomly sample an initial user. We then find the 20 closest users to this user in the graph embedding space, and randomly sample 5 from them. By sampling from users that are close to this initial user, we increase the likelihood that all six users will discuss similar topics, so focus areas can be more effective. But, we also don't only choose the closest five users every time, to encourage diversity, so the graph can potentially learn user similarity it doesn't already know.

When evaluating this, we see that for both down-



What is this user discussing and what is their perspective? Please summarize in one sentence.

Username: User 1
BIO: Email: ...; VERIFIED: 1; Follower count: 12345; Following Count: 678; Tweets Count: 901;
Some Tweets:
Tweet 0: The COVID-19 virus created fear across many people in both China and United States.
Tweet 1: ...
Tweet 2: ...
Summary: This user is discussing the COVID-19 virus, and how it caused many people to be afraid and that may have done more harm than the virus itself.

Figure 4: An example of the prompt we used to determine the user summary. For Twitter users, based on their bio, meta-data, and tweets, we create a summary. For Reddit, we use their comments.

Dataset: Model	**Coverage**	**# Samples**
Reddit Political: No Focus Areas	58.07	550
Reddit Political: Gold (Llama 2) Focus Areas	58.20	550
Reddit Political: LLM$_{\text{prompt}}$ Focus Areas: Supervised Learning	58.37	550
Reddit Political: LLM$_{\text{prompt}}$ Focus Areas: RL Curriculum Learning	**59.21**	550

Table 7: Results for when we use T5-Base as our LLM$_{\text{prompt}}$ and Llama 2 as our LLM$_{\text{task}}$. Results show improvements when using focus areas, a 1.96% performance increase. Although the improvements are not as significant as when we use ChatGPT as Llama is not a strong enough model to benefit from improved focus areas, we still see a strong improvement.

stream tasks, performance increases when this extra user community information is provided, but only when LLM$_{\text{task}}$ is used with focus areas and RL (our best model). Without focus areas, the communities formed by LLM$_{\text{task}}$ are likely incorrect, which is why performance drops, as incorrect user information is being spread in the graph. However, with focus areas, community detection performance improves (as we showed via other experiments where we had ground truth for this), and this leads to direct improvements in the downstream task. This is because, the Graph model leverages the user similarity in the newly formed communities, to determine which sources are likely to be fake news / politically biased, as this user information flows throughout the graph. The results in the main paper (Sec. 4.4.2) show the impact of forming good communities for downstream tasks.

D RL Training Details

In this section, we provide training details for our RL algorithm, and how exactly we train LLM$_{\text{prompt}}$ to work with LLM$_{\text{task}}$. We also discuss the mathematical details of our reward functions.

We initialize LLM$_{\text{prompt}}$ with an encoder-decoder T5-base, and supervised train it as a text generation task, using the gold focus areas as the training data. Specifically, if LLM$_{\text{prompt}}$ is parameterized by theta, we maximize $E[\log p_\theta(f|x)]$ where the goal is to generate focus areas f.

The second step of our training process is the RL stage, using different reward functions (detailed below). We continue training the policy network from the supervised learning stage (LLM$_{\text{prompt}}$), but now to maximize the reward from the reward functions, using the KL-regularized Proximal Policy Optimization (PPO) objective (Schulman et al., 2017). To do this, we sample batches, get the focus areas from LLM$_{\text{prompt}}$, pass them along with the user summaries to LLM$_{\text{task}}$, and then get user communities. We run the communities through our reward functions, compute the reward, and update LLM$_{\text{task}}$, by maximizing the PPO objective.

We now provide more details of RL4LMs, the public reinforcement learning library we used, as proposed by Ramamurthy et al. and used by Akyurek et al.. We provide an overview, more details can be found in Ramamurthy et al.. The RL4LMs library provides an OpenAI gym style API to allow us to easily train our models. In RL4LMs, each environment is viewed as a NLP task, i.e. generating focus areas from user summaries. Thus, there is a dataset $D = (x, y)$, where x is a language input (user summaries) and y is a target string (focus areas). Generation is viewed as a Markov Decision Process (MDP), consisting of states, actions, rewards, and transition functions. At each episode in the MDP, i.e. for a given dataset sample, the input x is provided to the model (LLM$_{\text{prompt}}$), and used as the initial state. An action a is then performed, which in this environment means to generate a token from the vocabulary. The transition function then models this and appends this action to the end of the state. This continues until the episode ends, i.e. when all the tokens are generated. At the end of an episode, a Reward based on the state and the gold focus area is provided. The environment can then be updated using the regularized KL reward, by training via PPO.

To improve the stability of training RL algorithms with NLP methods (i.e. handling large vocabulary sizes), RL4LMs also introduces NLPO: Natural Language PPO. NLPO maintains a masking policy, which is a copy of the current policy, but one that is updated only every u steps. This updating policy provides the original policy with an additional constraint that can help regularize the RL training.

D.1 Reward Function Details

Finally, we provide the mathematical details of each of the reward functions we used, expanding

Coverage: As outlined in Sec. 4.3.1, the goal of Coverage is to see how well LLM$_{\text{task}}$ can detect communities. Thus, we mathematically define it as:

$$\frac{\text{\# of correct pred.}}{\text{\# of correct} + \text{incorrect} + \text{missing pred.}} \quad (2)$$

Entity Frequency captures how may entities are being mentioned in the focus areas, that are useful for predicting the communities. This is motivated by the fact that good focus areas should mention detailed topics for LLM$_{\text{task}}$ to focus on. For simplicity, our goal is to have at least three useful entities in the focus areas. Mathematically, let g_e be number of entities mentioned more in one of the gold communities vs. another, and let f_e be the number of entities mentioned in the focus areas and in g_e. Then, the reward is: $min(1.0, f_e/3)$.

Focus Area Informativeness scores the focus areas using a pre-trained model from ChatGPT data. Thus, to compute the reward: Let LR be the Regression model scoring info and f be the focus area. Then, the reward is: $s_f = LR(f)$

Focus Area Length aims to make focus areas longer in length, so that they are potentially more detailed. To compute it, Let f_w be the number of words in the focus area, then the reward is: 0.5 if f_w < 10, 1.0 if f_w > 35, else $\frac{f_w - 10}{35 - 10} * (1.0 - 0.5) + 0.5$

E Curriculum Learning and Training Details

E.1 Training Details

We upload our entire code and data with this submission to make it easy to replicate our training process. Our code release also features all the hyper-parameters that we used to train our models. However, we additionally provide training details in this section. Upon acceptance, we will release all the code and data.

We train our T5-base model using the public repository published by Akyurek et al. and (Ramamurthy et al., 2022). Our models are trained using a 12GB Titan XP GPU card, and intial supervised training takes 1 day. Subsequently, future RL training iterations also take one day. We make calls to the OpenAI ChatGPT API, using the models available publicly in November 2023, at the time these experiments were performed. For Llama 2, we run a local model, with 70B parameters, published by the Llama-cpp-python library[4].

We used the development set to evaluate model performance, and choose the best hyper-parameters for our experiments.

As our prompt model, we train the T5-base model with a max prompt length of 650, for 120 epochs, a 0.00001 learning rate, and weight decay 0.01. For the RL stage, we fine-tune the T5-base model with all the same parameters, but a learning rate of 0.0001, entity coefficient of 0.1 and target KL of 3.

For downstream evaluation (news media profiling: news source political bias/factuality detection), our entire graph (train, test, and dev sets) has 2,969,854 edges, 81,326 nodes, 1,468 source nodes, and 35,099 user nodes.

E.2 Curriculum Learning

We use curriculum learning to learn our novel reward functions from Sec. 3.4.1. We do curriculum

[4] https://github.com/abetlen/llama-cpp-python

learning, as averaging the different rewards into one reward score, and using that one score throughout the training process, makes learning each reward difficult. This is because the model cannot separate between the rewards, as it only gets one score, so it can't learn each reward function individually, and performance suffers.

However, there are benefits to using multiple rewards, as evidenced in the RL literature (Dann et al., 2023). Particularly, in our case, we want focus areas to be informative, capture relevant entities, be detailed, and be useful for community detection. Thus, we designed our reward functions to capture this, so when we optimize these rewards, the focus areas have these properties. This leads to focus areas being useful for community detection, and without these rewards, they wouldn't be.

Using curriculum learning, we learn each reward function, one at a time, to ensure the model can optimize each one. We introduce an additional reward function once model performance does not improve on the validation set for three training iterations. We first optimize for downstream performance (coverage), and then entity frequency. While these rewards lead to our model producing useful focus areas with relevant entities, they are relatively short and not detailed enough to separate users into accurate communities. Thus, once performance doesn't increase on the validation set for three iterations, we add in the informativeness and finally length reward functions. As all the reward functions are added individually and used until performance stalls, they can be learned by the model, and they expand the initial focus areas to be more detailed and longer (thus also more useful). Once reward functions are used, they contribute equally to the final reward score (when compared to existing reward functions). However, as they are added sequentially, the model can still optimize them. We also use an additional reward, ROUGE score, which always contributes 25% to the final reward. This reward scores the generated focus areas using the ROUGE metric and the gold data, to make sure the model continues to generate focus areas that are grammatically sound.

In this way, curriculum learning helps us optimize all of our reward functions, learning focus areas that are useful for community detection. We additionally performed an ablation study on the individual reward functions in Sec. 5.1, which showed that while each reward function improves performance, learning them together through curriculum learning does the best.

F Gold Focus Area Generation

In this section, we discuss how we generate the gold focus areas to train $\text{LLM}_{\text{prompt}}$ in the initial supervised learning stage. To do this, we take advantage of the fact that we know the gold communities. We use LLM_{task} to generate the gold focus areas, as we hope to initialize our $\text{LLM}_{\text{prompt}}$ model to the performance of LLM_{task}. Further, using LLM_{task} to generate focus areas instead of humans allows us to quickly generate training data for a large amount of samples, which would otherwise be cost expensive.

Specifically, we prompt LLM_{task} to separate the communities given the user summaries. For this, we provide the users to LLM_{task} in sorted order (all users from first community first, all users from the second community second), asking it to provide the topics/entities to separate them. As LLM_{task} often generates extra text that should not be part of focus areas and also often mentions the ordering of the users (which will not be valid at test time since the users will be randomly ordered), we additionally provide extra instructions in the prompt to try and avoid this. The exact question we ask is shown in Fig. 5.

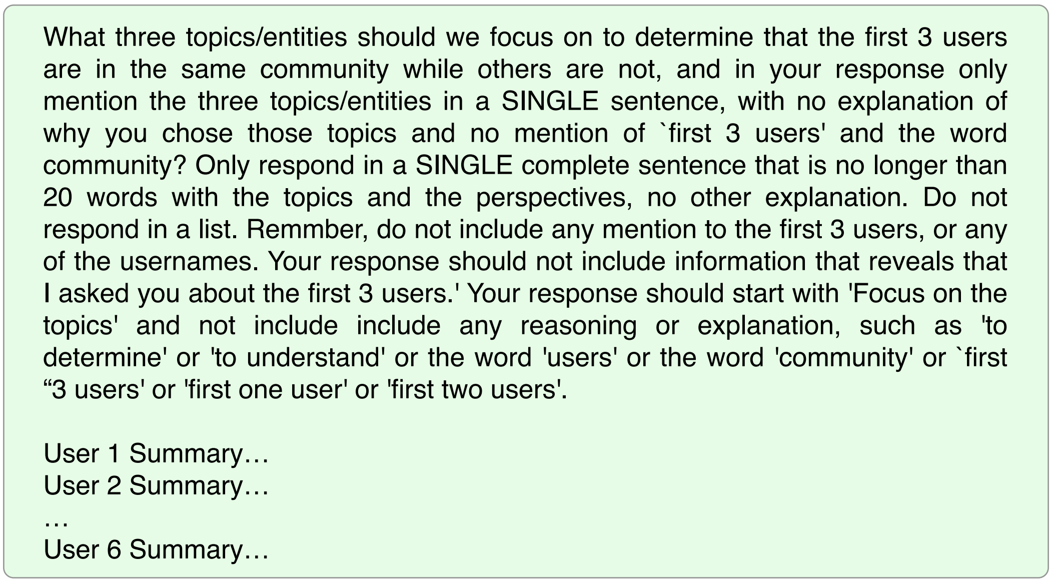
F.1 Llama 2 Results

In this section, we provide results for our models when using Llama 2 (Touvron et al., 2023) as LLM_{task}, instead of ChatGPT as used in the main paper. All other settings are the same as when we used ChatGPT. Results are shown in Table 7, and show similar trends to using ChatGPT, showing our framework generalizes across different LLM_{task} models. While the improvements of Llama 2 with focus areas are not as significant as ChatGPT, due to the fact that the Llama 2 model is not strong enough to take full advantage of focus areas, we still see significant improvements, showing the usefuleness of our framework.

G Discuss Cont: Human Analysis

In this section, we continue our discussion from Sec. 5.2 and provide more details of our human analysis process.

The goal of this step is to evaluate our focus areas, and determine if the focus areas generated by our framework are better than the ones produced by ChatGPT. While Sec. 4 shows that this is the case across a variety of community detection and



What three topics/entities should we focus on to determine that the first 3 users are in the same community while others are not, and in your response only mention the three topics/entities in a SINGLE sentence, with no explanation of why you chose those topics and no mention of `first 3 users' and the word community? Only respond in a SINGLE complete sentence that is no longer than 20 words with the topics and the perspectives, no other explanation. Do not respond in a list. Remmber, do not include any mention to the first 3 users, or any of the usernames. Your response should not include information that reveals that I asked you about the first 3 users.' Your response should start with 'Focus on the topics' and not include include any reasoning or explanation, such as 'to determine' or 'to understand' or the word 'users' or the word 'community' or `first “3 users' or 'first one user' or 'first two users'.

User 1 Summary…
User 2 Summary…
…
User 6 Summary…

Figure 5: An example of the prompt we used to generate gold focus areas. Given a set of six users, we ask LLM_{task} what makes the first three users part of the same community. We also add additional instructions to the prompt to make sure that the LLM responds only with focus areas, not extra information such as user ordering.

downstream tasks, in this section we have humans evaluate this.

To do this, we show three human annotators 50 samples (each human sees all 50). Each sample has one focus area from ChatGPT, and another generated by our best $\text{LLM}_{\text{prompt}}$ RL model for ChatGPT. For each sample, the human is asked to compare the focus areas, and then score them on a scale of 1-5, for grammatical correctness and usefulness. The usefulness rating identifies how useful the human believes the focus area will be to determine information communities. Ideally, a useful focus area should focus on divisive issues. The exact question we ask them is: Given two sentences (focus areas), score each on a scale of 1-5 (1 being lowest, 5 highest) for grammatical correctness and usefulness. The usefulness rating should capture how useful the focus area is to determine information communities. Ideally, a useful focus area should focus on divisive issues. The grammatical correctness rating should capture how grammatically correct the focus area is.

Results showed that while ChatGPT is more gramatically correct (4.95 vs 3.00), $\text{LLM}_{\text{prompt}}$ generates more useful focus areas (3.26 vs 3.07) across the 50 samples. This validates our experimental settings, where $\text{LLM}_{\text{prompt}}$'s focus areas lead to higher downstream performance, because they are more useful and focus the model on divisive issues to appropriately separate user communities.

The human annotators we used for this experiment were 20-30 year old male Ph.D. students in Computer Science and NLP, who are not authors of the paper or familiar with the study before the interaction process. One was Asian-American, one was Indian, and one was American. The students were provided fair working conditions and rewarded with research credit hours for their work in performing this annotation.

H Discussion: Case Study

In this section, we analyze our model by performing several case studies. We start by providing examples of high and low quality focus areas in Sec. H.1, making it clearer what we want our focus areas to looks like. Then, in Sec. H.2, we analyze the focus areas our trained model generates vs. ChatGPT, showing the benefit of our supervised and RL training procedure. Finally, in Sec. H.3 we show detailed examples of how focus areas improve community detection performance, showing snippets of user summaries and how the communities formed are better once focus areas are used.

H.1 High and Low Quality Focus Areas

We aim for focus areas to tell the bigger LLM, LLM_{task}, exactly what topics to focus on. Ideally, focus areas shouldn't be about high level issues, but rather divisive topics. In Table 8, we provide examples of high and low quality focus areas. All of these were generated by the LLM_{task} models presented in our framework. Note that the higher quality focus areas focus on issues, rather than just high level entities, which is what enables focus areas to

Low Quality Focus Areas	High Quality Focus Areas
Focus on Donald Trump.	Focus on Donald Trump's views on gun control.
Focus on political opinions and perspectives.	Focus on Democrats. Republican lawmakers and their perspectives. Republican lawmakers and their treatment of immigrant.
Focus on Fox News.	Focus on Sean Hannity's suitability as a diplomat.

Table 8: Examples of "high quality" and "low quality" focus areas, based on our definition. High quality focus areas tell the model what divisive issues/important topics and entities to focus on, so it can better detect the information community.

lead to better community detection. Moreover, they mention relevant entities, are informative, and are detailed, due the fact that we trained with several relevant reward functions (see Sec. 3.4.1).

H.2 ChatGPT vs LLM_{task} Focus Areas

Table. 9 shows several examples of focus areas generated by our $\text{LLM}_{\text{prompt}}$ model and ChatGPT. From this, we can see that our $\text{LLM}_{\text{prompt}}$ model generates more useful focus areas, as they inform LLM_{task} exactly of the topics and divisive issues to focus on to detect user information communities. This qualitatively shows the benefit of our Supervised + RL training procedure.

H.3 Focus Areas Improving Community Detection

Fig. 6 and Fig. 7 shows cases where focus areas can help improve community detection performance. On the contrary, Fig. 8 shows a case where focus areas can hurt community detection, if they are too specific (like in this case), or if they are too high-level/random (not shown).

I Discussion: Learned Communities for News Source Factuality Detection

In this section, we analyze how well LLM_{task} with focus areas allows us to learn communities that are relevant for the downstream task of news source factuality detection. To do this, we cluster (K-means, k=17) graph user embeddings before and after the LLM_{task} communities are added into the graph (as discussed in Sec. 4.1.3), and evaluate cluster purity.

Specifically, we clustered the users in the test set graph before and after LLM-based communities are created (i.e. before and after the new user edges based on the LLM_{task} communities are added to the graph), and evaluated the cluster purity. To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this is measured. We assign labels to the users by propagating directly downwards from the source factuality labels (i.e. a user that follows 3 high factuality sources and 1 tweets 1 low factuality article has a label "high" factuality). We cluster user graph embeddings, from the trained graph model, but do not do any training after the communities are created using the LLM.

The results show that user purity improves $\sim$ 3%, from 55.22 before to 58.66 after LLM_{task} communities are added to the graph, showing that the communities formed by the LLM are meaningful, as users with similar factuality labels cluster closer together.

J Discussion: Impact of RL

In this section, we further discuss the impact of the Reinforcement Learning (RL) stage on many of the results presented in the main paper, showing why this stage is crucical to both our community detection and downstream task performance.

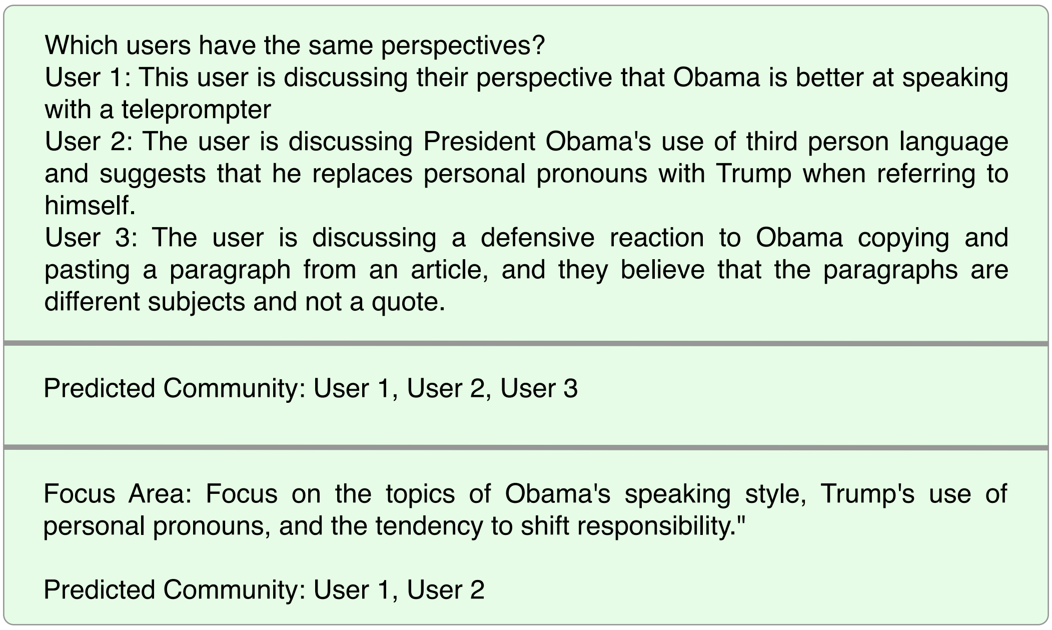
Specifically, when compared to the Supervised Learning approach when using ChatGPT as the LLM, Reddit Political improves from 45.48% to 47.85%, a > 5% relative performance improvement, and Reddit Economic improves from 44.00% to 45.58%, a > 3% relative performance improvement. All of this improvements is on unseen data from future time periods/topics, compared to the training set. We hypothesize that additional RL rewards, such as improving the grammar of the focus areas, could also improve performance more.

RL is also critical to our Downstream task evaluation on Fake News Detection and Political Bias Detection. Here, we compared to SOTA models that outperform multiple baselines (SVM, GNN, Trained Text Classifier, etc.). Our results show the benefit of building communities, and without the RL stage of our approach, this improvement would not be possible. The model without RL would perform worse than existing baselines on this downstream task.

Finally, the RL stage also leads to better focus

ChatGPT Focus Area	**$\text{LLM}_{\text{prompt}}$ Focus Area**
Focus on the Governor of Virginia's campaign ad.	Focus on social entitlement and equality for conservatives in America. diverse demographic in the White House.
Focus on clean coal.	Focus on death threats on Twitter. Free-Market Republicans. the entity Twitter and its
Focus on the Republican party.	Focus on Donald Trump and his perspective on the deal with North Korea. Donald Trump
Focus on political opinions and perspectives.	Focus on Democrats. Republican lawmakers and their perspectives. Republican lawmakers and their treatment of immigrant.
Focus on the topic of America and its current state.	Focus on Michael Savage's perspective on tying social media accounts to people's

Table 9: Examples of focus areas generated by ChatGPT and our best RL + Curriculum Learning $\text{LLM}_{\text{prompt}}$ model. The first section shows cases where the human annotator from Sec. 5.2 believed $\text{LLM}_{\text{prompt}}$ was better, and the second section where they rated ChatGPT to be better.



Which users have the same perspectives?
User 1: This user is discussing their perspective that Obama is better at speaking with a teleprompter
User 2: The user is discussing President Obama's use of third person language and suggests that he replaces personal pronouns with Trump when referring to himself.
User 3: The user is discussing a defensive reaction to Obama copying and pasting a paragraph from an article, and they believe that the paragraphs are different subjects and not a quote.

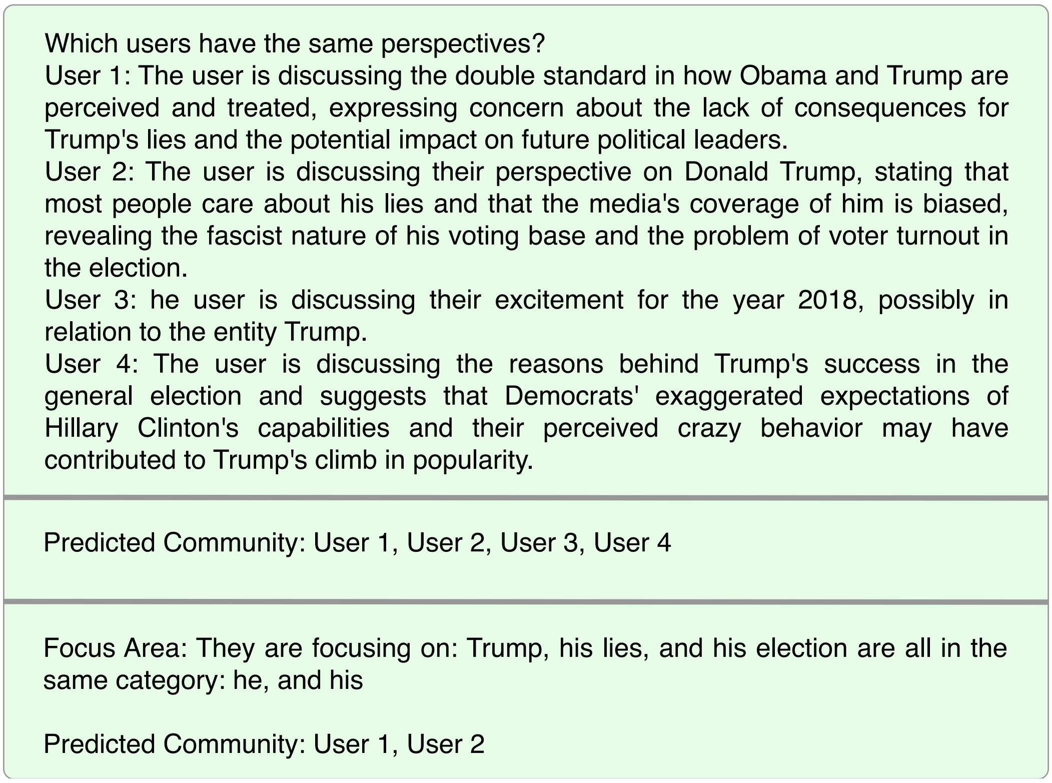
Predicted Community: User 1, User 2, User 3

Focus Area: Focus on the topics of Obama's speaking style, Trump's use of personal pronouns, and the tendency to shift responsibility."

Predicted Community: User 1, User 2

Figure 6: Success case: An example of how focus areas can improve community detection. We show a few users and a snippet of their summaries, in sorted order for clarity. Without focus areas, the LLM predicts that all three users should be in the same community, as they all discuss President Obama's speech. However, when asked to focus on Obama's speaking style by the focus area, the LLM correctly identifies that Users 1 and 2 are similar as they criticize Obama's speech, while User 3 is defensive of his speech.

areas (due to rewards like Focus Area Entities), which is important for the real-world deployment of our approach. Thus, RL is a critical component of our approach. Among other benefits, it leads to performance improvements and better focus areas.



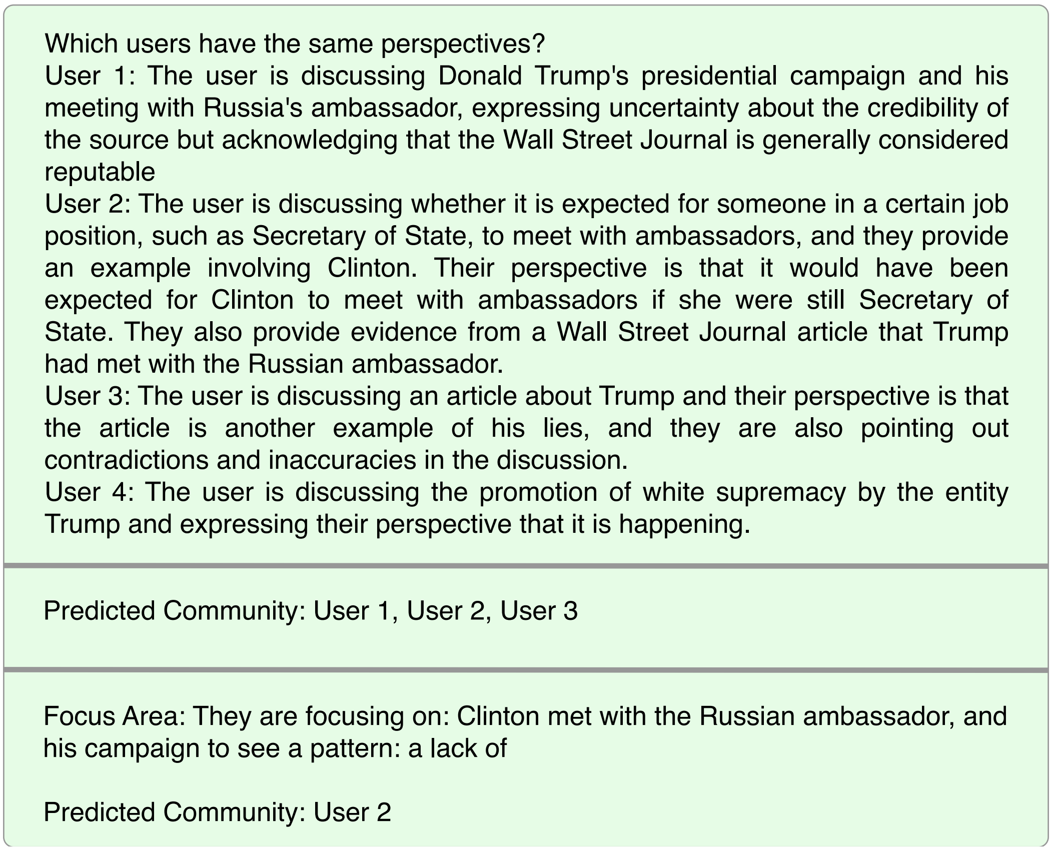
Which users have the same perspectives?
User 1: The user is discussing the double standard in how Obama and Trump are perceived and treated, expressing concern about the lack of consequences for Trump's lies and the potential impact on future political leaders.
User 2: The user is discussing their perspective on Donald Trump, stating that most people care about his lies and that the media's coverage of him is biased, revealing the fascist nature of his voting base and the problem of voter turnout in the election.
User 3: he user is discussing their excitement for the year 2018, possibly in relation to the entity Trump.
User 4: The user is discussing the reasons behind Trump's success in the general election and suggests that Democrats' exaggerated expectations of Hillary Clinton's capabilities and their perceived crazy behavior may have contributed to Trump's climb in popularity.

Predicted Community: User 1, User 2, User 3, User 4

Focus Area: They are focusing on: Trump, his lies, and his election are all in the same category: he, and his

Predicted Community: User 1, User 2

Figure 7: Success case: An example of how focus areas can improve community detection. We show a few users and a snippet of their summaries, in sorted order for clarity. Without focus areas, the LLM can't correctly predict the community, as they all discuss President Trump. However, when asked to focus on President Trump's lies, it's clear that the first two users are against Trump, and the LLM can predict it correctly.



Which users have the same perspectives?
User 1: The user is discussing Donald Trump's presidential campaign and his meeting with Russia's ambassador, expressing uncertainty about the credibility of the source but acknowledging that the Wall Street Journal is generally considered reputable
User 2: The user is discussing whether it is expected for someone in a certain job position, such as Secretary of State, to meet with ambassadors, and they provide an example involving Clinton. Their perspective is that it would have been expected for Clinton to meet with ambassadors if she were still Secretary of State. They also provide evidence from a Wall Street Journal article that Trump had met with the Russian ambassador.
User 3: The user is discussing an article about Trump and their perspective is that the article is another example of his lies, and they are also pointing out contradictions and inaccuracies in the discussion.
User 4: The user is discussing the promotion of white supremacy by the entity Trump and expressing their perspective that it is happening.

Predicted Community: User 1, User 2, User 3

Focus Area: They are focusing on: Clinton met with the Russian ambassador, and his campaign to see a pattern: a lack of

Predicted Community: User 2

Figure 8: Failure case: An example of how focus areas can hurt community detection. We show a few users and a snippet of their summaries, in sorted order for clarity. In this case, the focus area is too specific, leading to a one user community being formed, which is not very impactful.