Crowdsourcing annotations for harmful content classifiers

An update from GPAI's pilot project on political hate speech in India

October 2024

GPAI / THE GLOBAL PARTNERSHIP ON ARTIFICIAL INTELLIGENCE

Citation

Content Note

This report, given its subject of hate speech, contains quotes from content that many readers will find offensive. Of course, none of this quoted content in any way reflects the opinion of the report authors. Readers who may be disturbed by such content should please be advised.

Table of Contents

1. Introduction and Context

This report is a sequel to the [report we gave at last year's GPAI Summit in Delhi (GPAI, 2023)](#), that introduced our harmful content classification project and presented some initial results.

We begin in Section 2 by summarising the aims of the project, and the work described in our first report. In the remainder of the report, we present the new work we have done this year, and outline plans for future work.

2. Aim of the project: To explore a new paradigm in harmful content classification

Our project is about how social media platforms moderate 'harmful content' posted by users. Specifically, it's about the AI systems that are deployed in content moderation processes. The project is motivated by an analysis of problems with current company methods. These are introduced in GPAI (2023:1);[1] we will summarise them briefly here.

The central problem is that different companies operating in a given region *build their own private training sets* for the harmful content classifiers they use in content moderation in that region. This has three problematic consequences.

Firstly, companies implement *different definitions* of harmful content. This is partly because they choose different taxonomies of harmful content as starting points. Even if they choose similar content categories, their textual definitions of these categories are often subtly different, and these differences are likely to have consequences. The real definitions of the categories implemented by a classifier reside (collectively) in the labels assigned to items in its training set—and we don't have much information about what companies' training sets look like.

A second problem is precisely this lack of transparency about how training sets are created. Even if we put aside likely differences between classifiers, it is problematic that we *know very little* about how platforms' harmful content classifiers are trained. These classifiers are used to perform censorship, which involves difficult decisions about how to reconcile the protection of users' free speech with the need to keep harmful content off platforms. External stakeholders should know as much as possible about how these decisions are made.

A final problem with the status quo is that it is inefficient for companies to build their own training sets for harmful content classifiers. Larger training sets are better, in general, for any machine learning task. If companies pooled their resources and created a single training set, which they all trained on, we can expect improvements in performance for all platforms—especially in regions where resources are lacking. There are additional benefits with this scenario, if companies' classifiers are all *evaluated* on (a held-out portion of) this pooled dataset. Firstly, the evaluation would be in the form that is standard for all supervised learning in AI: performance on held-out training data. (Companies currently report a different figure, the 'proactive detection rate'; see e.g. Meta, 2024. This measure is certainly useful in its own right, but it is no substitute for the standard

---

[1] Numbers in our citations of last year's report refer to sections of that report.

metric used throughout AI.) Secondly, companies could be *directly compared* in the performance of their classifiers, which would create useful competition between companies. This competitive scenario is also a standard component of AI research: machine learning conferences have been organised around 'shared tasks' of this kind for the last 20 years at least (see e.g. Voorhees and Harman, 2005).

Our project on harmful content aims to pilot the scenario just sketched, where a single training set of harmful content items is created in a given region, for use by all social media platforms operating in that region. The idea is to compile this training set by *consulting a representative sample of the public* in that same region. The basic proposal is to apply a simple democratic principle: representative members of the public will be given content items from the training set, and asked to provide suitable 'labels' for these items. Of course, this cannot be done for all forms of harmful content. But for harmful content in political domains, we feel it may offer an interesting way of choosing the labels on which classifiers are trained. It essentially amounts to running an *opinion poll* on each item in the training set. Naturally, we can expect a great deal of disagreement amongst annotators (just as we would in any political domain). But disagreement can be made to play a useful role in training classifiers. Rather than training a classifier to return the 'majority label' for a given item, we can readily train it to return a *probability distribution* over possible labels, reflecting the true distribution recorded from annotators (see e.g. Uma et al., 2021). Probability distributions tell us about the level of disagreement between annotators about a given item. We suggest this knowledge can be helpful in adjusting moderation actions—for instance, by moderating more leniently if disagreement is high, to keep active debates alive. These proposals are all described in more detail in GPAI (2023:2).

A key aim in our project is to build classifiers whose decisions have a maximum of accountability. Having training sets assembled through 'democratic' processes provides some measure of accountability. But it's also important to confront the trade-offs between free speech and harm prevention in the curation of the training set. In our project, we use a taxonomy for harmful content that directly addresses this trade-off. Rather than defining categories of content items semantically, we define them *operationally*, by the moderation actions that they require. Our categorical taxonomy comprises four actions: 'remove', 'downrank', 'leave untouched', and 'uprank'. For items to be downranked or upranked, we have a second, continuous measure of harm (or good), relating to how much downranking (or upranking) should happen. These operational definitions force annotators to confront the tradeoff between free speech preservation and platform safety in every annotation they perform. A classifier that is trained on such annotations will incorporate annotators' judgements about the importance of free speech, played off against their judgements of item harmfulness—again, sampled democratically from the population. Using this process should give the trained classifier's decisions an additional measure of accountability. A graphical summary of the scheme we are piloting is shown in Figure 2.1. This figure is intended as a preview: details of these proposals are described more fully in GPAI (2023:3).

Social Media Governance: Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India

2
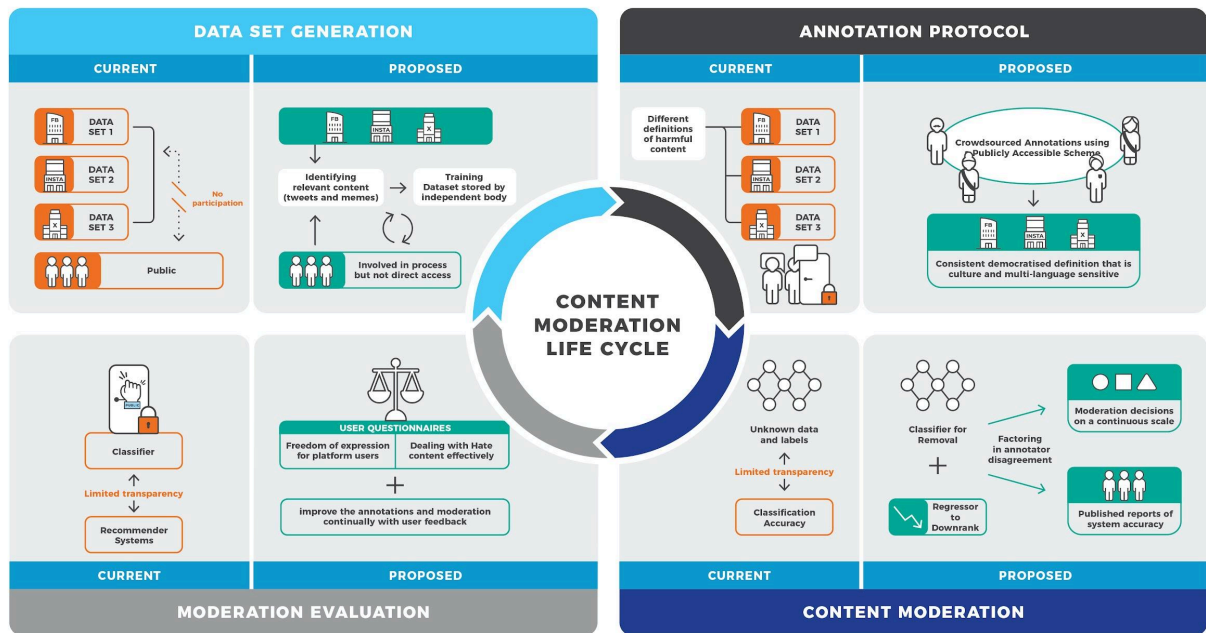
Figure 2.1: Graphical Summary of the Piloted Content Moderation Scheme and Contrasts with the Status Quo

Social Media Governance: Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India

3

3. Our pilot study on political hate speech in India

Of course the proposals set out above need to be tested. It might simply not be possible to use judgements from 'the crowd' to create a workable content classifier. We have been running a small pilot study to try out our proposed scheme for training a political hate speech classifier. Our pilot runs in India, GPAI's Chair nation for 2024. India is a country where political hate speech has particularly serious consequences in real life, frequently triggering actual violence (see e.g. Mirchanandi, 2018). It is also the world's largest democracy, so trials of new democratic methodologies in AI are of particular interest. In this report, we present the methods and results from our pilot study so far. The work we present here extends the initial trials we described in GPAI (2023:4-6), and contributes to a growing literature on hate speech detection in India (see e.g. Kumar et al., 2018; Chakravarthi et al., 2020; Saroj and Pal, 2020; Dowlagar and Ramidi, 2021; Romim et al., 2021; Das et al., 2024; Mandl et al., 2019; 2020; 2021).

Our work has explored two *forms* of political content: **Tweets** (short passages of text), and **memes** (images, often containing textual captions). These both feature widely in Indian political discussions carried out on social media. We will describe our dataset of Tweets in Section 4, and our dataset of memes in Section 5 (with some additional discussion about the importance of memes as a medium). The sets of annotators we used for Tweets and memes have some overlap, but we will present them separately, in these same sections.

For both Tweets and memes, our analyses focus on content gathered in the run-up to Indian elections. We consider two electoral contexts: the general elections, also known as **Lok Sabha** elections, and the state elections, referred to as **Vidhan Sabha** elections. The Lok Sabha elections determine the composition of the lower house of the Indian Parliament, where MPs are elected to represent the diverse constituencies across the country. These elections happen every five years. Vidhan Sabha elections are held at the state level to elect members to state legislative assemblies. Each state in India has its own Vidhan Sabha, and these elections occur periodically, determining the governance of individual states. For further details, see GPAI (2023:6.4).

Our work has also made use of two forms of annotation, that pick up on the two types of 'operational' decision proposed in Section 2. In **discrete** annotation studies, annotators assign each item a discrete label (from the set 'remove', 'downrank', 'leave untouched' and 'uprank'). These annotations will serve to train a *classifier*, whose job is to make a discrete decision between these alternative actions. In **continuous** annotation studies, annotations are used to place training items at points on a continuous scale of 'hatefulness'. These annotations will serve to train a *regression system*, that is to be used for items that were identified by the classifier as 'downrank' or 'uprank', and will determine how much the item is up- or down-ranked. In our work on Tweets, we have conducted a discrete annotation study (described in Sections 6 and 7) and a continuous annotation study (described in Section 8). Our work on memes only features a discrete annotation study so far: we describe this in Section 9. In sum, we report three annotation studies: two for Tweets, and one for memes.

Our project also explores two ways of *assigning* content items to annotators. In a **dense** assignment protocol, each annotator receives the same set of items to annotate (or the same set of annotation tasks). In a **sparse** protocol, annotators are assigned items at random. This latter method is called

'sparse' because it is designed for larger datasets, where it's not feasible to ask annotators to look at every item, or perform every annotation task. (We are interested in exploring how our study can be scaled up to larger datasets and annotator groups.) Our Tweet annotation studies in GPAI (2023) were all 'dense'. This year, our discrete Tweet annotation study used a 'sparse' assignment. Our continuous Tweet study and our meme study explore both 'sparse' and 'dense' assignment methods.

For each annotation study, we have developed a customised user interface for annotators, and tailored methods for choosing items to present to annotators in the interface. The interface for discrete annotation studies presents items one-by-one for labelling. The interface for continuous annotation studies presents *in pairs*, and requires them to select which item in each pair is 'most hateful'. We will introduce these interfaces at appropriate points in Sections 6–9.

We examine the results of annotation studies in a variety of ways. In all cases, we analyse the amount of disagreement we find between annotators. For Tweets, we use additional methods for discrete and continuous annotation studies. For the discrete study, we explore how useful the assembled dataset is in training (or rather fine-tuning) a text classifier to identify the relevant categories. The classification study is described in Section 7. For the continuous study, our first task is to convert the set of pairwise judgements from annotators into valuations of each individual Tweet on a continuous scale of 'hatefulness', for use in down- or up-ranking. For this task, we use two varieties of the Bradley-Terry model (see Firth, 2005; Caron and Doucet, 2012), which we describe in Sections 8.3 and 8.4.

Before we conclude, we'll note a couple of caveats with the study we present here: one primarily technical, one primarily legal.

3.1.   A pilot project of the proposed method

We should emphasise that the annotation studies we report here are *pilots* of the crowdsourced exercise we have in mind. There are small numbers of annotators. Our focus is on developing the materials and methods that could be used in a much larger study: namely the annotation platform, and the methods for analysing annotations. We are also interested in conducting 'reality checks' on the results of the annotation studies, to see if they are likely to be able to deliver the kind of training which is needed.

3.2.   Legal definitions of harmful content, and the HASOC dataset

It's also important to note that many jurisdictions around the world have *legal* definitions of harmful content that need to be considered, alongside the judgements of citizens. At present, these legal definitions all concern categories of content that platforms must *remove*. For instance, there are laws requiring 'hate speech' or 'bullying' or 'threats of violence' to be removed (see e.g. Paz et al., 2020; El Asam and Samara, 2016; Murphy, 2019). 'Hate speech' is often defined as directing hate towards a defined *group* of people, rather than an individual. Bullying and threats can be directed towards individuals, and need not count as hate speech (though they can also be hate speech).

Of course, legal definitions of content that must be removed supersede the definitions delivered by the kind of crowdsourcing exercise we are exploring: citizens can't take the law into their own hands. But crowdsourcing can still play an important role alongside legal definitions, because there

are many domains where the law does not apply, but moderation is still very important. Social media platforms' content moderation policies necessarily go beyond the law in several respects. For one thing, the law does not currently engage with moderation actions like downranking: partly because they are very new, and the law does not yet intervene at this level in platform operation, and partly because the law can't yet articulate the relevant distinctions with enough subtlety. More broadly, black-letter law can only make rather *general* provisions about categories of content, because it is expressed as linguistic generalisations. Where further detail is needed about a particular situation, the law sometimes relies on a body of case law, expressing a large dataset of decisions by individual judges, and sometimes on the decisions of ordinary people in juries. The crowdsourcing scheme we are exploring can be thought of as another possible way of extending legally defined categories with the collective decisions of citizens. For instance, we could refine our current pilot by asking citizens to judge whether a given content item is an instance of 'hate speech' or 'threats of violence'—again, with provisions for cases of borderline content. In this case, annotators would be playing a role analogous to jurors, asked to assess the facts of specific cases.

Some harmful content classification exercises pay attention to legal categories in this way. In particular, the HASOC annotation scheme (Mandl et al., 2020) defines 'hate speech' as directed towards groups, and 'offensive speech' as directed towards individuals. (We'll make some use of this scheme in our own work on training classifiers; we will introduce the scheme in more detail in Section 7.1).

4.   Our dataset of Tweets, and our set of Tweet annotators

4.1.   The Tweet dataset

We use a standard method for gathering Tweets, which is to retrieve Tweets with relevant hashtags and relevant posters, to create a 'broad dataset', and then applying various filtering operations to orient the dataset towards relevant political discussions. Our dataset of Tweets is based on the set we used last year: the relevant hashtags and filters are described in GPAI (2023:6.4).

In our GPAI (2023) study, we gathered two datasets: one for the 2019 Lok Sabha (national) elections (607 Tweets in total, grouped into three 'subsets'), and one for the 2022 Vidhan Sabha (state) elections (400 Tweets): again, see GPAI (2023:6.4) for details of this process. In this year's study, we expanded on the 2019 Lok Sabha dataset, adding a further 333 Tweets to the first 'subset' of data (to give a total of 1340 Tweets). The additions were prompted by our discovering some further relevant tags and posters. Specifically, upon analysing the entire dataset (more than 90,000 Tweets collected in the context of the Lok Sabha Elections 2019), we identified a very relevant frequently occurring hashtag ('#'): *#ElectionCommission*. The Election Commission of India (ECI) is a constitutional body established by the Constitution of India empowered to conduct free and fair elections in India. We extracted all the Tweets with this hashtag (2172 Tweets) and added them to our broad Lok Sabha dataset. We then reran our filtering process over this extended dataset, using a method similar to last year.

The complete process of assembling our Lok Sabha dataset can be described as follows. First, we identified the most frequently occurring tags ('@') in the whole dataset. Figure 1 shows the number of Tweets for a set of relevant tags, including the most common ones. The figure includes the *@ECISVEEP* tag—the official Twitter handle of the ECI. We then selected Tweets containing the most frequent four tags—namely, the two most prominent political parties (BJP and Congress), and the main leader of each party—and also the *@ECISVEEP* tag. We then extracted all the Tweets out of the 2172 Tweets containing these five tags. The final size of our Lok Sabha Tweet dataset is 790.
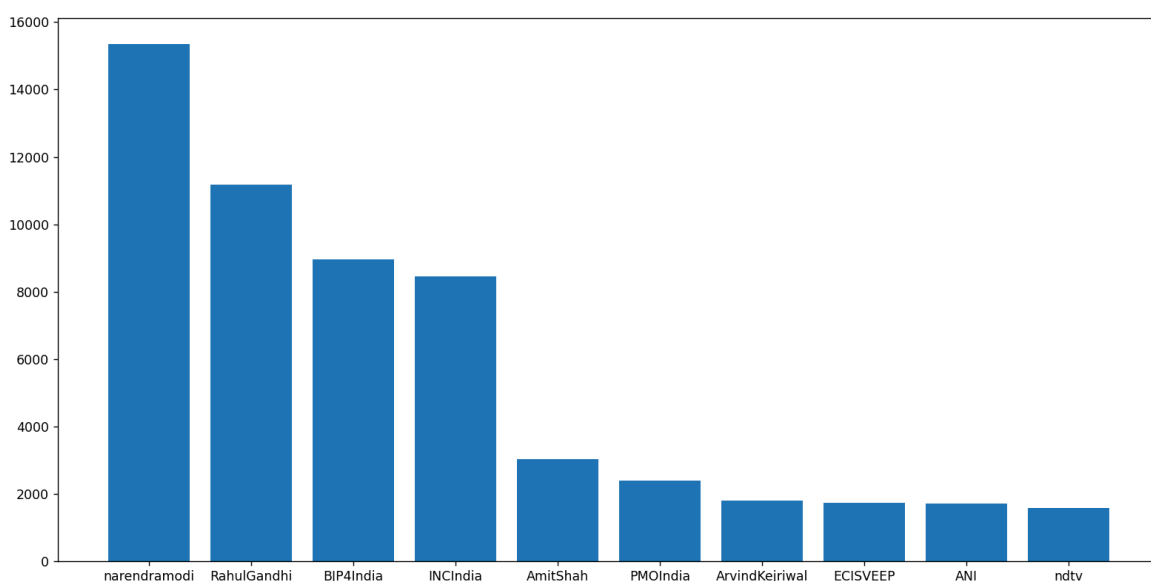
Figure 1: Analysis of the most frequently occurring tags in our 'broad' Lok Sabha 2019 dataset

4.2. Tweet annotators

We used 10 annotators in GPAI (2023). In this year's discrete Tweet annotation study, for the additional 333 Lok Sabha Tweets, we extended the number of annotators to 17. As before, our main concern was to pick a set of annotators from a range of different ages, genders, educational levels, career positions, ethnicities, religions and geographic regions. Demographics of the 17 annotators who participated in the pilot study are shown in Table 1. Annotator names are redacted for anonymity.

Annotator. ID	Age	Sex	Education	Career Status	Ethnicity	Religion	State
A1	22	M	BE	Student	General	Hinduism	West Bengal
A2	21	F	BTech	Student	General	Hinduism	West Bengal
A3	24	M	BSc	Student	SC	Hinduism	West Bengal
A4	29	F	MTech	Mid Career IT Professional	General	Hinduism	West Bengal
A5	45	M	BSc	Mid Career Content/ Digital Marketeer	General	Hinduism	Punjab
A6	46	M	BE	Business	General	Hinduism	Karnataka/UttarPradesh
A7	22	F	BTech	Student	SC	Hinduism	West Bengal
A8	20−30	M	Mtech	Mid career IT professional	General	Hinduism	India/USA
A9	22	M	BE	Student	General	Hinduism	Bihar
A10	20−30	M	Bachelors	Software Engineer	General	Islam	Karnataka
A11	20−30	M	Bachelors	Tech support	General	Christian	Karnataka
A12	22	F	BA	Student	General	Hinduism	West Bengal
A13	30−40	M	Masters	MBA	General	Hinduism	Haryana
A14	25	F	Masters	Digital Marketer	General	Hinduism	West Bengal
A15	23	M	Masters	Student	SC	Hinduism	West Bengal
A16	20−30	M	BTech	Software Engineer	OBC	Islam	West Bengal
A17	40−50	M	Masters	Service Engineer	General	Hinduism	West Bengal

Table 1. Summary demographics for the 17 annotators in the discrete Tweet study

In this year's continuous Tweet annotation study (which was new), we used 25 annotators. Demographics for these annotators, who partially overlap with those used for the discrete task, are shown in Table 2.

Annotator. ID	Age	Sex	Education	Career Status	Ethnicity	Religion	State
A1	22	M	BE	Student	General	Hinduism	West Bengal
A2	21	F	BTech	Student	General	Hinduism	West Bengal
A3	24	M	BSc	Student	SC	Hinduism	West Bengal
A4	29	F	MTech	Mid Career IT Professional	General	Hinduism	West Bengal
A5	45	M	BSc	Mid Career Content/ Digital Marketeer	General	Hinduism	Punjab
A6	46	M	BE	Business	General	Hinduism	Karnataka/Uttar Pradesh
A7	22	F	BTech	Student	SC	Hinduism	West Bengal
A8	20−30	M	MTech	Mid career IT professional	General	Hinduism	India/USA
A9	25	F	BCA	IT professional	OBC	Islam	West Bengal
A10	22	M	BE	Student	General	Hinduism	Bihar
A11	20−30	M	Bachelors	Software Engineer	General	Islam	Karnatak
A12	50−60	F	Bachelors	Digital Marketer	SC	Hinduism	Karnataka
A13	45	M	MCA	Software developer	SC	Hinduism	Punjab
A14	20−30	M	Bachelors	Tech support	General	Christian	Karnataka
A15	22	F	BA	Student	General	Hinduism	West Bengal
A16	20−30	M	Bachelors	Software Developer	OBC	Islam	West Bengal
A17	30−40	M	Masters	MBA	General	Hinduism	Haryana
A18	25	F	Masters	Digital Marketer	General	Christian	Goa
A19	32	F	Bachelors	Housewife	General	Hinduism	West Bengal
A20	23	M	Masters	Student	SC	Hinduism	West Bengal
A21	20−30	F	BTech	Software Engineer	OBC	Islam	West Bengal
A22	24	M	BTech	Software Engineer	SC	Hinduism	Bihar
A23	40−50	M	Masters	Service Engineer	General	Hinduism	West Bengal

Annot ator. ID	Age	Sex	Education	Career Status	Ethnicity	Religion	State
A24	30–40	M	Masters	Operations Manager	General	Hinduism	Punjab
A25	30	F	Bachelors	Software Developer	General	Hinduism	West Bengal

Table 2. Summary demographics for the 25 annotators in the continuous Tweet study

5.   Our dataset of memes, and our set of meme annotators

5.1.   Why memes?

In recent years, the rise of memes on social media platforms like Facebook, X/Twitter, and Instagram has garnered significant attention due to their widespread influence and ability to shape public discourse. While often humorous, many memes employ sarcasm and dark humor to propagate societal harm. Consequently, meme analysis is crucial for identifying offensive content and analyzing psychological responses. However, detecting offensiveness in memes poses a significant challenge for automated models. This difficulty arises from the relatively weak correlation between their textual and visual components, further complicated by contextual nuances, subcultural references, and subjective interpretations.

Recently, religious tensions in West Bengal, the state in which our project is based, have increased due to internet content, which shapes public opinion and leads to real-world actions. The socio-cultural history of West Bengal, marked by significant political riots and the 1947 partition of India, provides a backdrop for examining its current social and cultural landscape. In this context, memes created by networked groups often spread threats in previously peaceful areas. These memes focus on topics like the perceived danger to Hindus, fear of Bangladeshi infiltrators, and defamation of the Prophet, uniting individuals against a perceived 'other.' Political communication has shifted from a top-down approach to a participatory media, where memes are used to shape knowledge and influence bio-politics. People often fall for memes tied to religious beliefs and blind faith, embodying the 'virus of the mind'—a powerful force that paralyzes rational thought for effective propagation.

As a first step towards identifying hate speech in online memes, we have created a new meme dataset in the Indian political context by searching the Web. We scraped Facebook and Instagram groups on Indian Politics and then removed the ones which are not in the Indian political context. We also searched on Twitter by following the *trending* political hashtags related to the Lok Sabha Elections of 2024 which took place from 19th April to 1st June 2024.à

5.2.   The meme dataset

As a first step in our analysis of memes, we created a preliminary meme dataset in the Indian political context of 2024. Some examples are shown in Figure 2. Our method for gathering this dataset was more informal: we relied on informants who are knowledgeable about memes, and Indian politics. These informants scraped Facebook and Instagram groups on Indian Politics and then removed the ones which are not in the Indian political context. They also searched on Twitter

by following trending political hashtags related to the Lok Sabha Elections of 2024 which took place from 19th April to 1st June 2024.
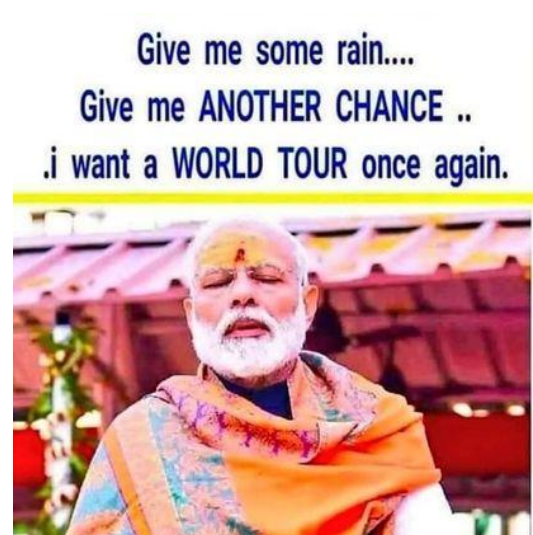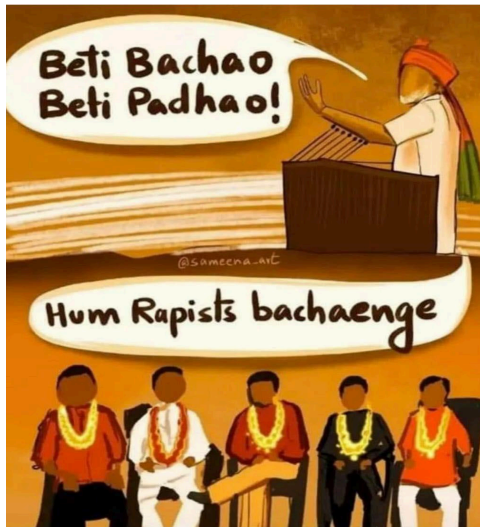
Figure 2. Examples of memes in our preliminary meme dataset.

5.3.   Meme annotators

The set of annotators for our meme study is shown in Table 3. As already noted, this group of annotators overlaps to some extent with the annotators for our Tweets study.

Annotator. ID	Age	Sex	Education	Career Status	Ethnicity	Religion	State
A1	22	M	BE	Student	General	Hinduism	West Bengal
A2	21	F	BTech	Student	General	Hinduism	West Bengal

Social Media Governance: Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India

11

Annot ator. ID	Age	Sex	Education	Career Status	Ethnicity	Religion	State
A3	24	M	BSc	Student	SC	Hinduism	West Bengal
A4	29	F	MTech	Mid Career IT Professional	General	Hinduism	West Bengal
A5	45	M	BSc	Mid Career Content/ Digital Marketeer	General	Hinduism	Punjab
A6	46	M	BE	Business	General	Hinduism	Karnataka/Ut tar Pradesh
A7	22	F	BTech	Student	SC	Hinduism	West Bengal
A8	20−30	M	MTech	Mid career IT professional	General	Hinduism	India/USA
A9	22	M	BE	Student	General	Hinduism	Bihar
A10	20−30	M	Bachelors	Software Engineer	General	Islam	Karnataka
A11	50−60	F	Bachelors	Digital Marketer	SC	Hinduism	Karnataka
A12	20−30	M	Bachelors	Tech support	General	Christian	Karnataka
A13	22	F	BA	Student	General	Hinduism	West Bengal
A14	20-30	M	Bachelors	Software Developer	OBC	Islam	West Bengal
A15	30−40	M	Masters	MBA	General	Hinduism	Haryana
A16	25	F	Masters	Digital Marketer	General	Hinduism	West Bengal
A17	23	M	Masters	Student	SC	Hinduism	West Bengal
A18	20−30	F	BTech	Software Engineer	OBC	Islam	West Bengal
A19	40−50	M	Masters	Service Engineer	General	Hinduism	West Bengal
A20	30−40	M	Masters	Operations Manager	General	Hinduism	Punjab
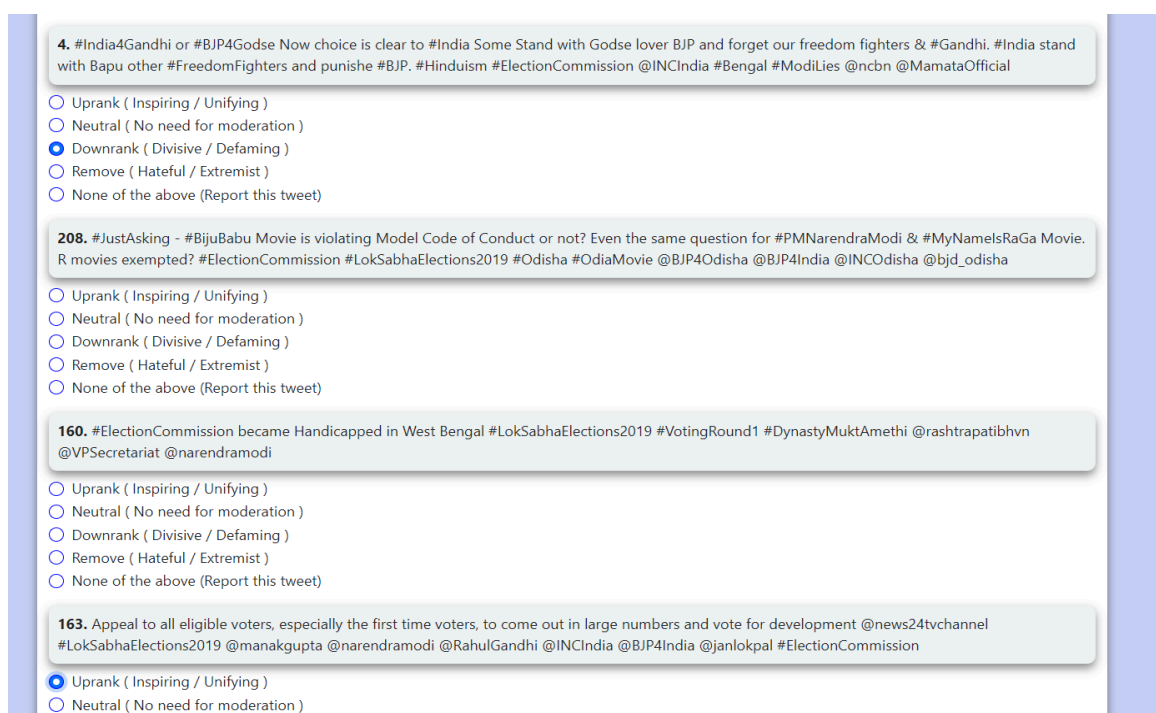
Table 3. Summary demographics for the 20 annotators in the Memes study

6.   Our discrete annotation study of Tweets

6.1.   An upgraded annotation platform for 'sparse' annotations

The annotation platform we developed last year, for 'dense' discrete Tweet annotations, is described in GPAI (2023:6.2). This year, for the 'sparse' Tweet annotation study, on the 333 newly-gathered Lok Sabha Tweets, we created a separate interface, with a separate server. The key difference in this new interface is that the user can choose to ignore content items they do not want to annotate.

The new interface can be found at http://annotate.infomaticae.com/v2/. Screenshots are shown in Figures 3-5. Note that as before, there is an option for annotators to 'report' Tweets—which means to indicate they are unclassifiable given the classes they are provided.

Figure 3: Tweet annotation form for the discrete study
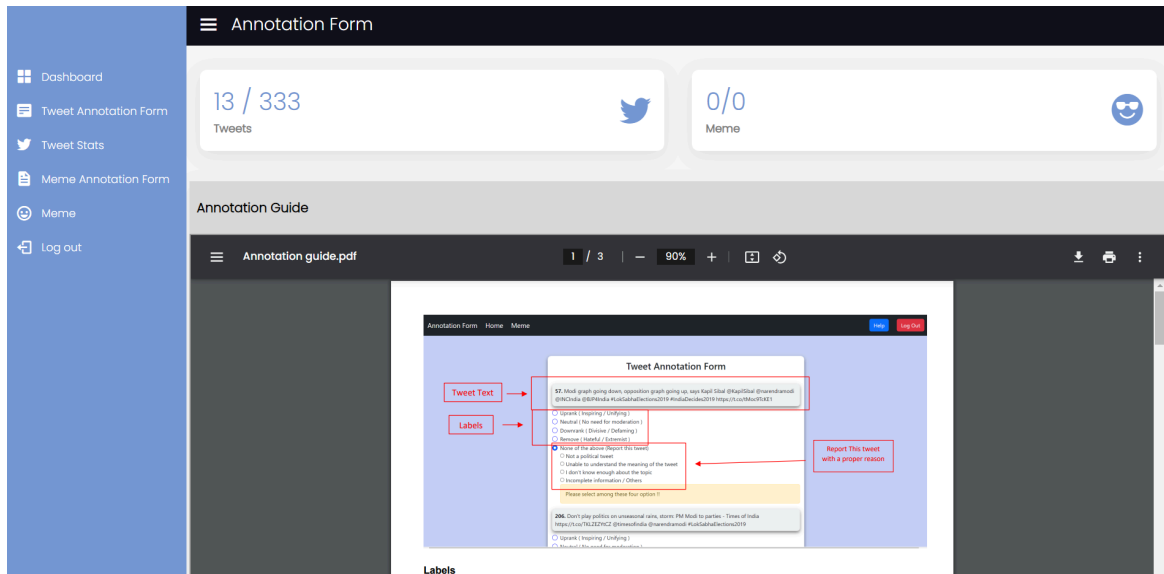
Figure 4: User dashboard for the discrete study
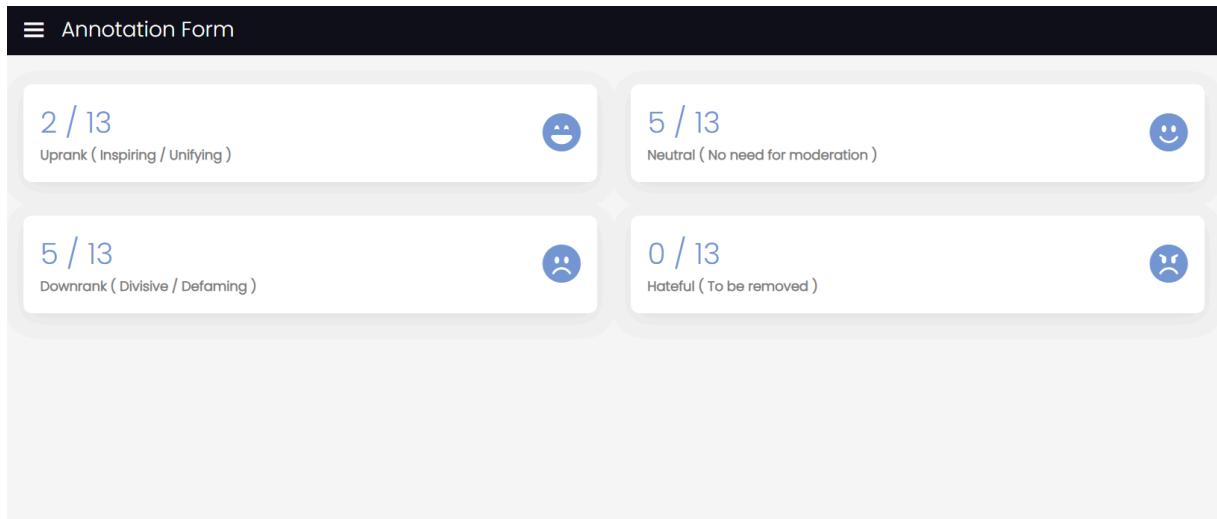
Figure 5: Tweet stats of a user by label class for the discrete study

6.2.   An analysis of data in the 'sparse' annotation study

The extended portion of our training dataset, containing 333 Tweets, was annotated by 17 annotators, as already noted. In this section, we will provide an analysis focusing on the disagreement that was found between annotators.

Since we are using a sparse assignment method for this study, we first report the distribution of the number of annotations per Tweet. This is shown in Figure 6.
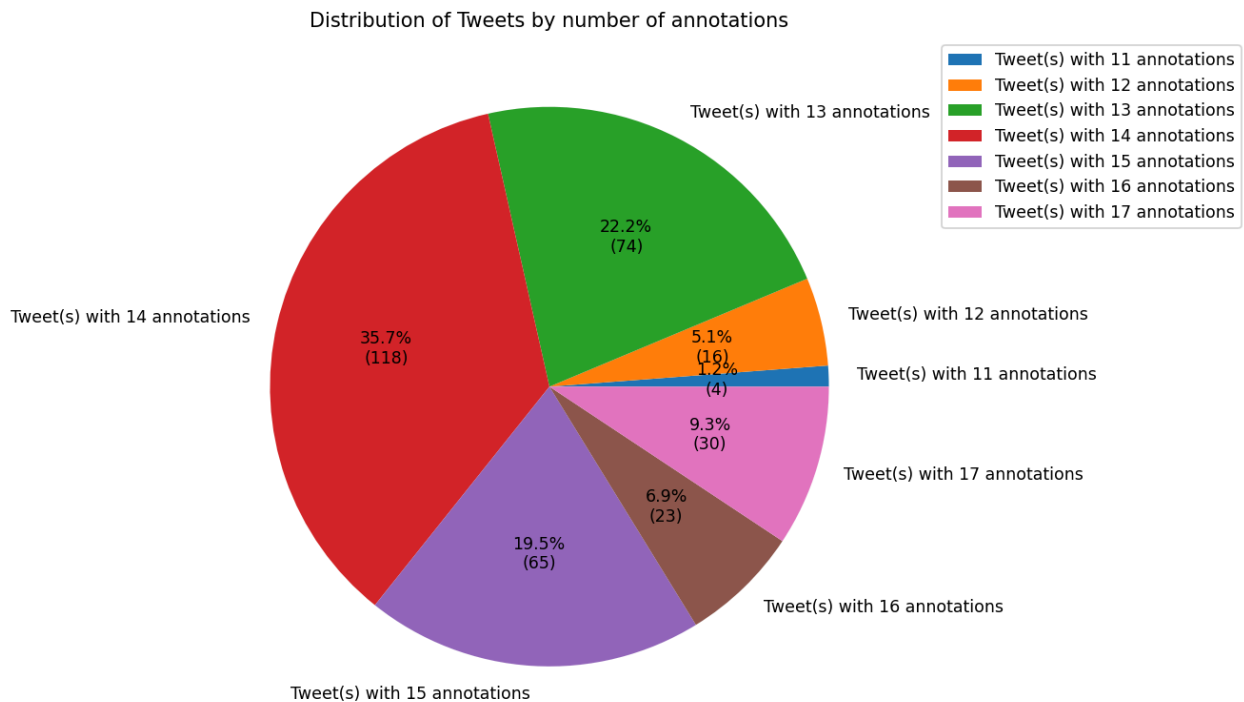
Distribution of Tweets by number of annotations

As shown in Figure 6, each Tweet was annotated by at least 11 annotators, with the majority of Tweets receiving more than 11 annotations, and even up to 17 annotations (which indicates that every annotator contributed to these annotations).

For analysis purposes, we first separate out the Tweets that were 'reported' (i.e., flagged as unclassifiable) by a majority of annotators. There were 15 of these. The remaining Tweets were further analysed into three discrete classes, based on the amount of agreement between the annotators:

- ○ Unanimous Agreement (39 Tweets)
- ○ Disagreement by a degree of 1 (36 Tweets)
- ○ Disagreement by a degree of 2 (134 Tweets)
- ○ Disagreement by a degree of 3 (109 Tweets).

These results are depicted in Figure 7.
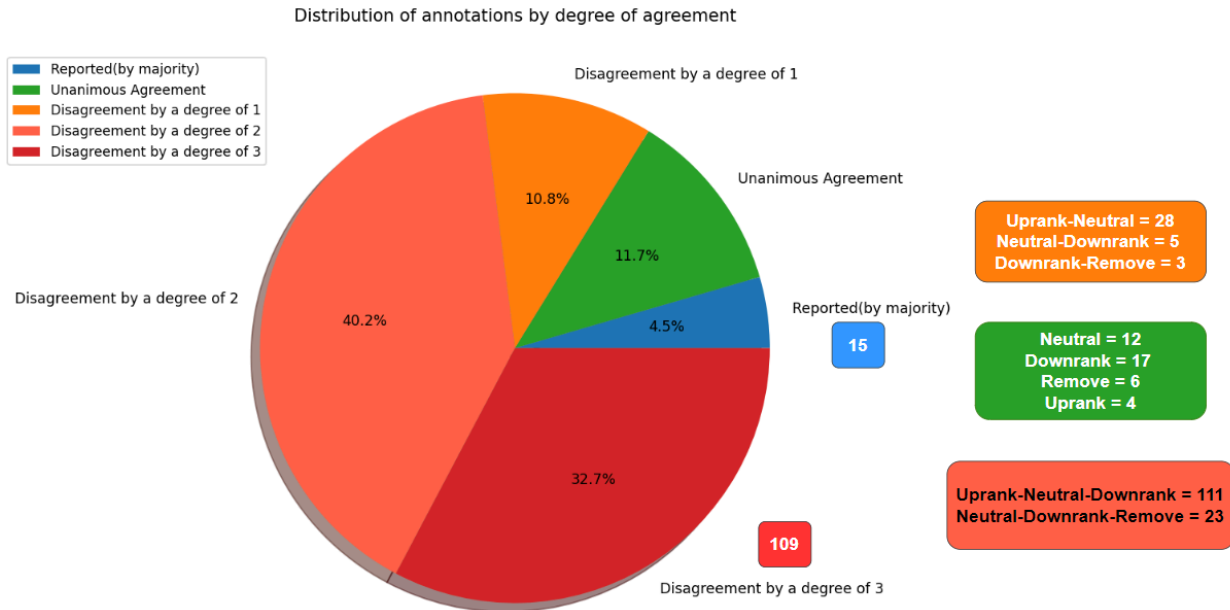
Figure 7: Distribution of sparse Tweet annotations by degree of disagreement

Figure 7 also displays the distribution of Tweets across categories of "Unanimous Agreement", "Disagreement by a degree of 1", and "Disagreement by a degree of 2", based on their corresponding class labels.

As we can observe, there are varying degrees of disagreement across the dataset—including a major portion of Tweets for which there is significant disagreement. This result is not unexpected across a team of annotators, but it is useful in providing preliminary evidence that the amount of disagreement will vary significantly over content items.

Finally, we report an entropy analysis of the annotations for this dataset. Figure 8 shows a histogram of entropy ranges for the dataset. The Tweets with most disagreement shown in Figure 7 have entropies in the highest entropy ranges in Figure 8. This analysis further suggests that a moderate to high level of annotator disagreement is common.

Histogram of Entropy for Sentiment Annotations
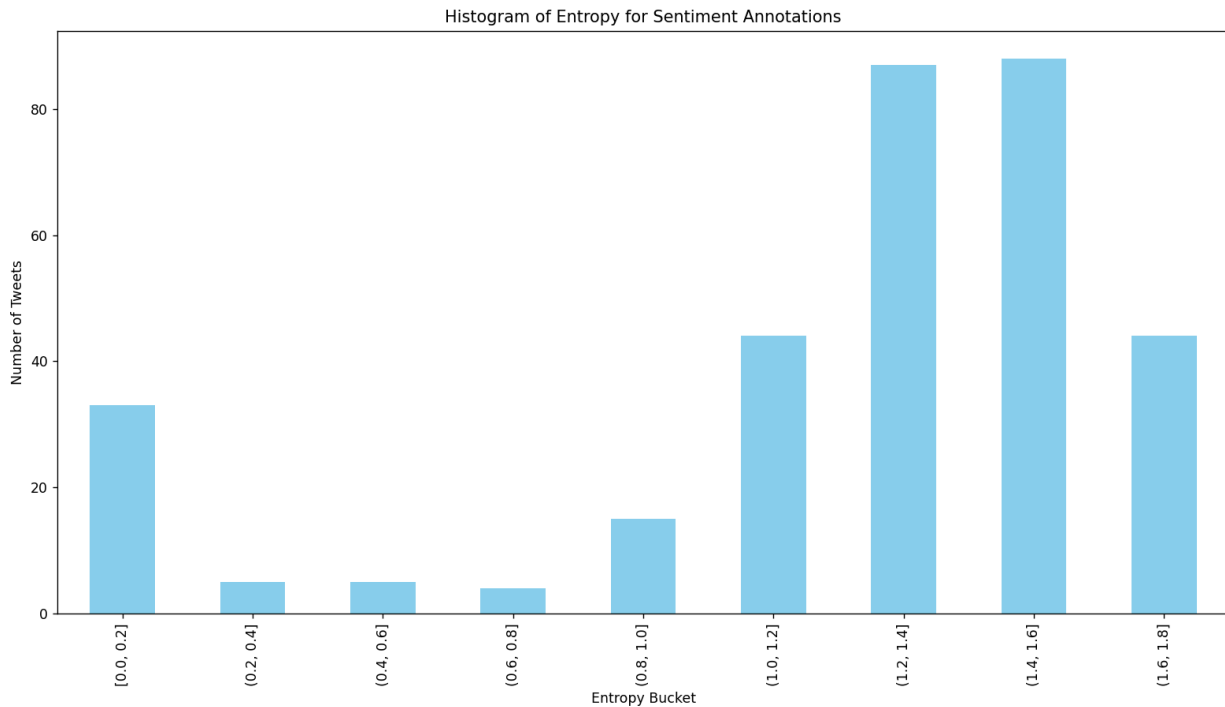
Figure 8: Histogram of entropy for sparse Tweet annotations

7. Using data from the discrete Tweet annotation study to train a text classifier

In this section, we explore how the dataset of discrete annotations we have gathered in our project so far can be used to train—or rather, fine-tune—a text classifier. The best performing text classification models at present are all built using transformer-based 'large language models' (LLMs), and we will focus on LLMs in our study. LLMs first need to be pre-trained on a large corpus of text from the relevant Indian context. For this purpose, we make use of the HASOC dataset: we describe this dataset in Section 7.1. There are various special characteristics of our own dataset of Tweets, and also of the HASOC database, relating to the multilinguality and code-switching that are found in an Indian context: we introduce these in Section 7.2. In Section 7.3, we introduce our experiments with three LLM-based classifiers, trained on HASOC, and fine-tuned on our discrete Tweet annotations. In Section 7.4, we present the results of these experiments.

7.1. The HASOC dataset, for initial training

The HASOC (Hate Speech and Offensive Content) dataset (Mandl et al., 2019; 2020; Modha et al., 2021) is a collection of text data used for training and evaluating models for hate speech and offensive content detection in Indo-European languages. At present, it is the primary resource for studying these topics in the open, academic domain: an extremely valuable resource for researchers working on hate speech detection in multilingual contexts.

The dataset is designed specifically for the *HASOC shared task*, an annual competition that evaluates systems for identifying hateful content across Indo-European languages (Mandl et al., 2019; 2020; 2021). The key characteristics of the dataset are as follows.

First, it has a multilingual focus, on a suitable mixture of languages. The HASOC dataset includes data in multiple Indo-European languages, such as English, Hindi, Marathi, etc. This allows researchers to develop and evaluate models that can handle hate speech across different languages.

Second, there is a focus on 'real-world' online language data. The HASOC dataset mostly comprises data scraped from social media platforms like Twitter, reflecting the types of content encountered in online environments. The dataset was gathered during India's severe second wave of COVID-19: in this period, social media discussions were significantly influenced by pandemic-related topics. Tweets regarding the post-poll violence in West Bengal were also included in the collection.

7.1.1.   HASOC's annotation schemes

Parts of the HASOC dataset have already been annotated for hate speech and offensive content, in valuable prior work (see again Mandl et al., 2019; 2020; Modha et al., 2021). As already noted in Section 3.2, the annotation schemes used in HASOC are somewhat different to the 'operational' categories we are using in our project ('remove', 'downrank', 'neutral', 'uprank', as described in Section 3). In this section, we'll introduce HASOC's annotation schemes. The differences between our annotation scheme and the HASOC scheme will likely create problems for our fine-tuning process—but we have to start with the data that's available for us.

HASOC's annotation schemes make reference to categories of 'hate speech', 'offensive content' and 'profane content'. In the HASOC scheme, 'hateful' content must contain hate towards an identified *group* of people. 'Offensive' content is harmful content directed to an individual that doesn't express hate towards a broader group. It would include individual bullying and harassment; also threats of violence. (All these conceptions also appear in legislation in various forms, as we discussed in Section 3.2; they are often echoed in platform content policies.) 'Profane' content, meanwhile, just contains profane language. This can occur in many contexts that aren't hateful or harmful; it's useful to identify profane content independently, to help isolate cases where it co-occurs with hate or offensiveness.

Drawing on these definitions, HASOC uses two annotation schemes, of different granularities (see e.g. Mandl et al., 2020). HASOC's 'Subtask A' uses a coarse-grained binary scheme, with two categories: 'Hateful, Offensive or Profane' (HOF), and 'Not Hateful, Offensive or Profane' (NOT). 'Subtask B' further divides the HOF category into its components: 'Hateful' content, 'Offensive' content, and 'Profane' content.

HASOC had its own process for selecting Tweets to annotate. The first step was to deploy a simple preliminary classifier trained elsewhere, to identify a set of candidate HOF and NOT (non-HOF) Tweets. This classifier used a SVM model on *N*-gram features (Mandl et al., 2021). Tweets identified by the classifier as HOF and non-HOF were randomly selected for more detailed annotation. (An additional 5% of non-HOF Tweets were included to ensure a balanced dataset.) Profane keywords were used to further balance the dataset. Tweets were annotated by at least two annotators, with conflicts resolved by a third annotator. (HASOC annotations provide a single 'right answer', unlike our soft-labels approach.)

7.2.   Multilinguality issues in our dataset and the HASOC dataset, and other caveats

Like the HASOC dataset, the Tweets in our dataset are in a mixture of languages. India is a multilingual country, and switching between languages is common. Our dataset is primarily English and Hindi. The HASOC dataset is more diverse, which is likely to pose further problems in fine-tuning classifiers on our dataset—but again, this is the data we have to work with.

It's also useful to note that utterances in both our dataset and the HASOC dataset frequently switch between languages *within a single phrase or message*. We will refer to this as 'code-switching'. Code-switching can introduce subtle semantic nuances that might be missed by models trained on monolingual data. These issues with multilinguality present particular problems for hate speech classifiers; but they are important problems to confront. There are ways of approaching them using explicit language-detection mechanisms, combined with several monolingual LLMs. But our approach is to train a single LLM, in the distinctive mixture of languages found in the dataset.

As an aside, it's worth noting that all language processing models can inherit biases present in their training data. This can lead to discriminatory outcomes in hate speech detection. Additionally, the opaque nature of some deep learning models can make it difficult to understand how they arrive at their decisions, hindering trust and transparency.

7.3.   Our experiments with three LLM classifiers

We have chosen three deep-learning based transformer models—RoBERTa (Liu et al., 2019), ALBERT (Lan, 2019) and DistilBERT (Sanh et al., 2019)—to obtain some preliminary results for our newly gathered dataset of discrete annotations. As already noted, our protocol is to pre-train each model on the HASOC dataset, and then fine-tune it on our own (much smaller) dataset. In fact, there are two pre-training steps, because each LLM is already pre-trained on a very large language corpus, through a process described in the paper that presents it. We begin by loading the model's pre-trained weights. Starting from these weights, we then further train the full model on the HASOC datasets from 2020 and 2021 (a total of 3709 + 3844 = 7553 Tweets), using their coarser-grained 'Subtask A' annotations, which distinguish 'HOF' content from non-HOF content (see Section 7.1). Finally, we fine-tune the full model on our dataset of discrete annotations.

We describe our experiments with RoBERTa, ALBERT and DistilBERT in Sections 7.3.1–7.3.3. In each case, we introduce the base LLM and its architecture, then report the results of our training/fine-tuning process. In this latter step, we report the result of training on HASOC alone as a baseline, alongside the result of training on HASOC and then fine-tuning on our dataset. In the former case, we evaluate on 20% of held-out data from the HASOC dataset. In the latter, we pretrain on the full HASOC dataset, then fine-tune on 80% of our own dataset, and evaluate on the remaining 20%.

Recall that there are four labels in our discrete annotations dataset: 'remove', 'downrank', 'neutral', and 'uprank'. In our fine-tuning experiments, we present results for training/testing on these four categories; we also present results for a coarser-grained training/testing process

using only two categories: one combining 'downrank' and 'remove'; the other combining 'neutral' and 'uprank'.

7.3.1. The RoBERTa model

RoBERTa (Robustly Optimized BERT Pretraining Approach) (Liu, 2019) is a transformer-based model designed to enhance the performance of the BERT model (Devlin et al., 2019) by optimizing its pretraining process. RoBERTa retains the underlying architecture of BERT but modifies key aspects of the training procedure, including using larger datasets and longer training times, and adjustments to hyperparameters. RoBERTa was motivated by the observation that BERT's pretraining setup was not fully optimized, and more extensive training, coupled with removal of the Next Sentence Prediction (NSP) objective, could significantly improve the model's overall performance on downstream NLP tasks.

The primary contributions of RoBERTa lie in its robust pretraining strategy. By employing a larger and more diverse corpus, training the model with longer sequences, and using dynamic masking, RoBERTa consistently outperforms BERT across various NLP benchmarks.

RoBERTa's architecture

RoBERTa's architecture is shown in Figure 9 (courtesy of Huang et al. , 2021).
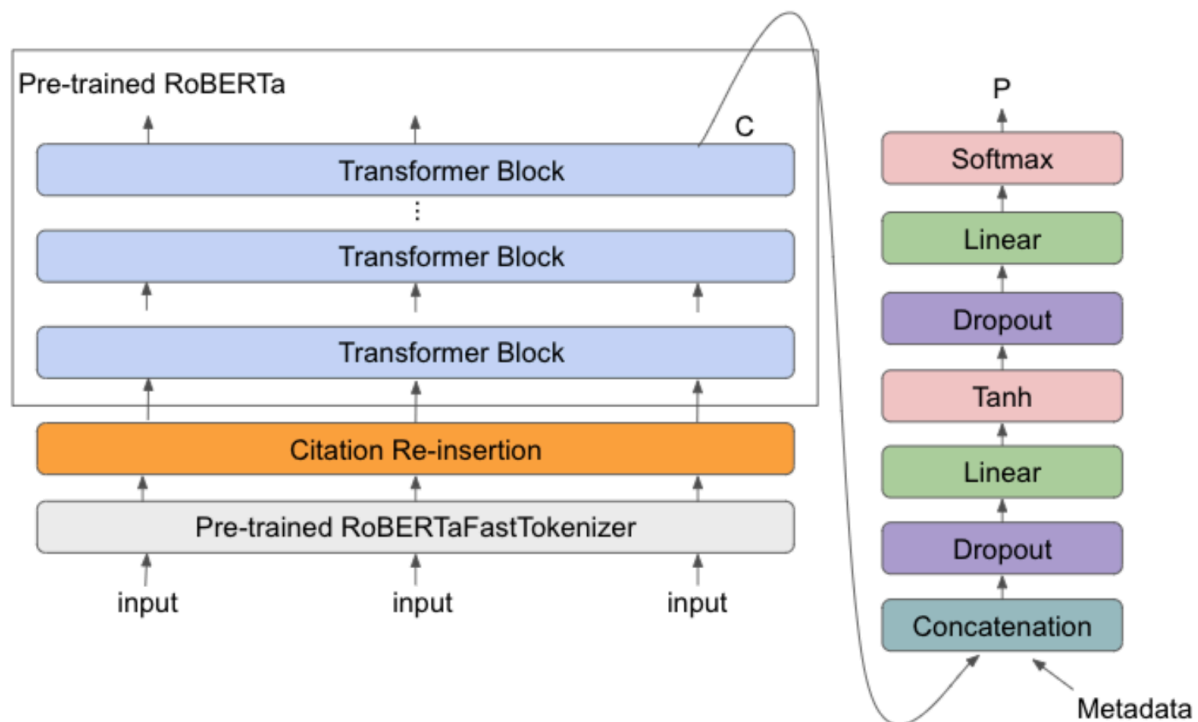
Figure 9. Architecture of the RoBERTa language model

The main components of RoBERTa's architecture are:

1. **Input Embeddings:** RoBERTa uses WordPiece tokenization to break down input text into smaller subwords or tokens. Each token is then converted into a vector representation by summing token embeddings, position embeddings (to maintain word order), and segment embeddings. Unlike BERT, RoBERTa does not utilize the Next Sentence Prediction (NSP) task, so it omits segment embeddings related to sentence pairs.
2. **Transformer Encoder**: RoBERTa has a stack of transformer encoder layers, each consisting of two key components:
   a. Self-Attention Mechanism: This mechanism allows the model to attend to different parts of the input sentence to capture dependencies between tokens, regardless of their distance.
   b. Feed-Forward Neural Networks: After self-attention, RoBERTa applies a feed-forward neural network (FFN) to each position separately, followed by layer normalization and residual connections.
3. **Output Layer:** The final hidden state from the last transformer layer is passed to a task-specific layer, depending on the NLP task. For classification tasks, for example, a simple linear layer with a softmax function is used to generate probabilities over the class labels.
4. **Dynamic Masking:** One of the major innovations in RoBERTa's pretraining process is dynamic masking, where the masked tokens change across different epochs of training. This ensures that the model learns from a wider variety of masked tokens, enhancing the quality of learned representations.

Results of training/fine-tuning with RoBERTa

Training and Evaluation Setup:

- Number of epochs : 10
- Batch Size (Training and Evaluation) : 32
- Weight decay : 0.01

Performance Metrics – (trained on HASOC data - tested on HASOC data)

- **Accuracy :** 0.8596
- **Precision :** 0.9168
- **Recall   :** 0.8859
- **F1 Score :** 0.8796

Performance Metrics for 4 labels – (trained on HASOC data + our data - tested on our data - 80:20 split)

- **Accuracy :** 0.7842
- **Precision :** 0.8729
- **Recall   :** 0.8213
- **F1 Score :** 0.7677

Performance Metrics for 2 labels – (trained on HASOC data + our data - tested on our data - 80:20 split)

- **Accuracy :** 0.8151
- **Precision :** 0.8529
- **Recall   :** 0.8017
- **F1 Score :** 0.8477

7.3.2.   The ALBERT model

ALBERT (A Lite BERT for Self-Supervised Learning of Language Representations; Lan, 2019) is a transformer-based model that aims to reduce the memory and computational demands of BERT (Devlin et al., 2019) while maintaining competitive performance. ALBERT introduces key modifications to the BERT architecture to make it more efficient in terms of both model size and training time, without sacrificing accuracy on downstream tasks.

The primary contributions of ALBERT are twofold: first, it introduces **parameter reduction techniques** that significantly decrease the number of parameters, allowing for a more scalable model; second, it incorporates a **sentence-order prediction (SOP)** task to improve the model's understanding of inter-sentence coherence, replacing the Next Sentence Prediction (NSP) task used in BERT.

ALBERT's architecture

ALBERT's architecture is shown in Figure 10 (courtesy of Zhou et al., 2022).
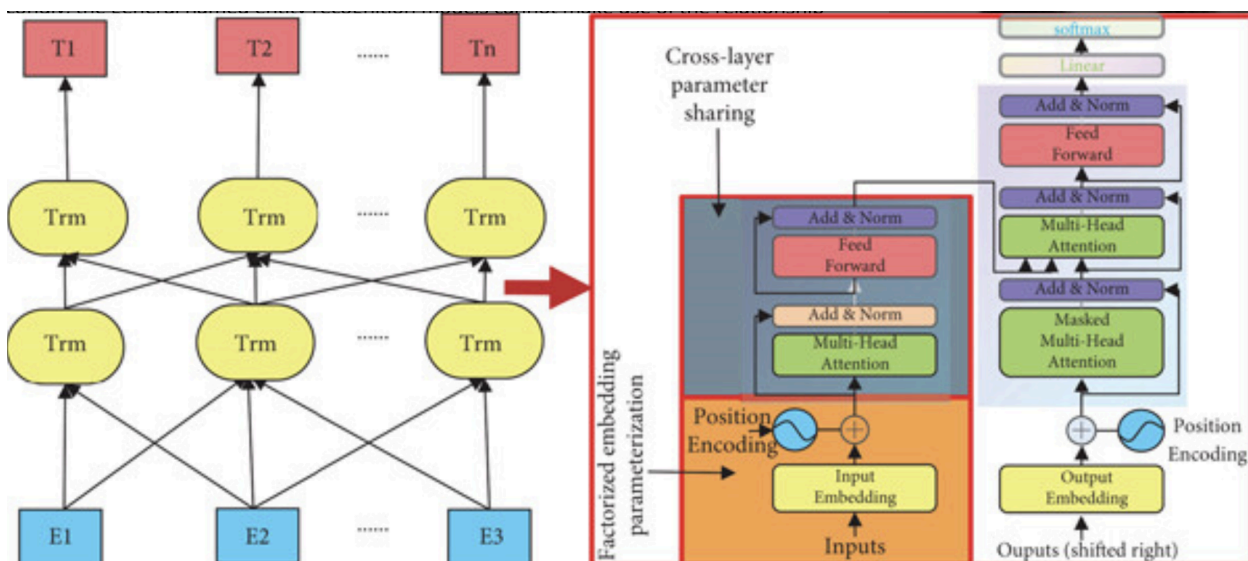
Figure 10. Architecture of the ALBERT language model

1. **Factorized Embedding Parameterization**: One of the major changes in ALBERT is the **factorized embedding** layer, which decouples the size of the hidden layer from the size of the vocabulary embeddings. In BERT, the embedding matrix grows significantly as the hidden dimension increases, leading to a large number of parameters. ALBERT addresses this by using a smaller embedding size (e.g. 128 dimensions) to represent tokens, while the hidden layer retains a larger dimension (e.g., 768 or 1024). This factorization reduces the

number of parameters in the embedding layer while still allowing for a large hidden layer, saving memory and computational cost without degrading performance.

2. **Cross-Layer Parameter Sharing**:
To further reduce the number of parameters, ALBERT introduces **parameter sharing** across layers. Instead of learning separate parameters for each layer, ALBERT shares weights across all transformer layers. This drastically cuts down the number of parameters, especially in deeper models. Two types of parameter sharing are implemented:
   a. **All-parameters sharing**: All layers share the same parameters, including feed-forward and attention weights.
   b. **Attention-sharing**: Only the attention weights are shared between layers.

3. **Sentence-Order Prediction (SOP) Task**:
ALBERT replaces BERT's Next Sentence Prediction (NSP) task with the **Sentence-Order Prediction (SOP)** task during pretraining. In the SOP task, the model is presented with two consecutive sentences from a document, and it must predict whether the sentences are in the correct order or if they have been swapped. This task emphasizes sentence-level coherence and helps the model better capture dependencies between sentences, an area where NSP often falls short. SOP improves performance on tasks that require understanding relationships between sentences, such as natural language inference and reading comprehension.

4. **Transformer Layers**:
Similar to BERT, ALBERT uses a stack of transformer layers, each consisting of self-attention and feed-forward networks. The number of layers and attention heads are configurable depending on the version of ALBERT. Although the architecture remains similar to BERT, the use of parameter sharing and factorized embedding ensures that ALBERT is much more parameter-efficient while still benefiting from the depth of transformer layers.

5. **Dynamic Masking and Pretraining**:
ALBERT uses the masked language modeling (MLM) objective similar to BERT, where 15% of the tokens in the input sequence are randomly masked, and the model is trained to predict the masked tokens. Dynamic masking ensures that different tokens are masked during each epoch, helping the model generalize better across different portions of the input.

Results of training/fine-tuning with ALBERT

Training and Evaluation Setup:

- Number of epochs : 10
- Batch Size (Training and Evaluation) : 32
- Weight decay : 0.01

Performance Metrics - (trained on HASOC data - tested on HASOC data)

- **Accuracy :** 0.8871
- **Precision :** 0.8562
- **Recall :** 0.8254
- **F1 Score :** 0.8732

Performance Metrics for 4 labels - (trained on HASOC data + our data - tested on our data - 80:20 split)

- **Accuracy :** 0.7408
- **Precision :** 0.7221
- **Recall   :** 0.6915
- **F1 Score :** 0.6872

Performance Metrics for 2 labels - (trained on HASOC data + our data - tested on our data - 80:20 split)

- **Accuracy :** 0.7862
- **Precision :** 0.7523
- **Recall   :** 0.7011
- **F1 Score :** 0.7527

7.3.3.   The DistilBERT model

DistilBERT (Sanh et al., 2019) is a compact version of the BERT (Devlin et al., 2019) model, developed by Hugging Face. It retains 97% of BERT's language understanding while being 60% faster and 40% smaller. This makes it an efficient alternative for deploying NLP models in resource-constrained environments.

DistilBERT's architecture

The architecture of DistilBERT is shown in Figure 11 (courtesy of Adel et al., 2022).
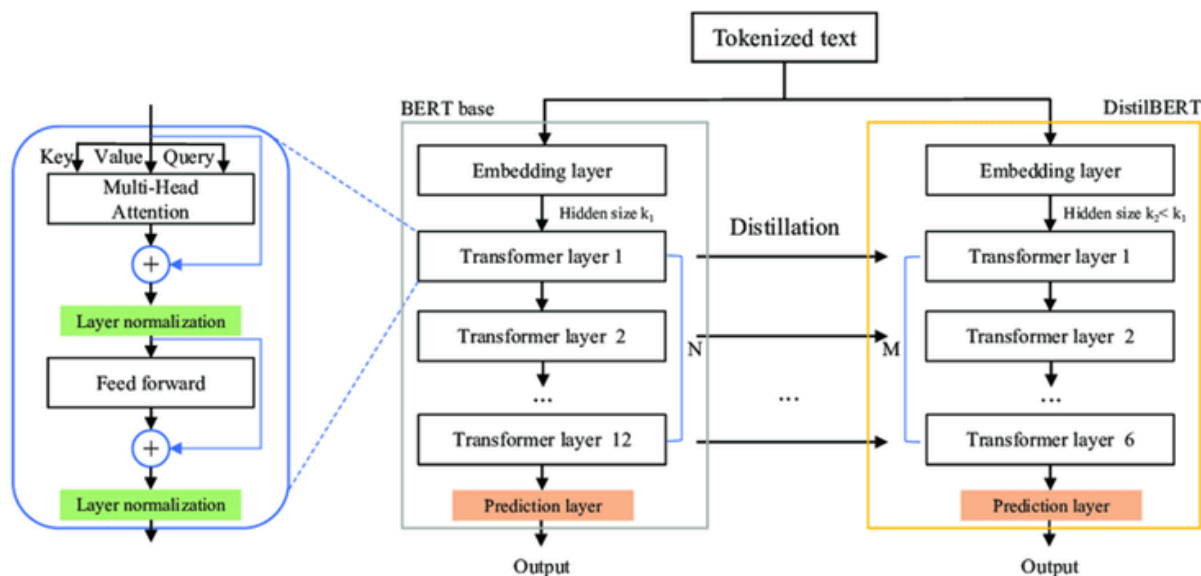

Figure 11. The architecture of the DistilBERT model

DistilBERT is designed to distill the knowledge of BERT into a smaller model. The main components of DistilBERT's architecture are:

1. **Input Embeddings:** Similar to BERT, DistilBERT uses embeddings to convert input tokens into continuous vector representations, including token embeddings, segment embeddings, and position embeddings.
2. **Transformer Encoder**: Consists of multiple layers of self-attention and feed-forward neural networks. DistilBERT uses the same Transformer encoder architecture as BERT but with fewer layers.
   a. Multi-Head Self-Attention: Allows the model to focus on different parts of the input simultaneously.
   b. Feed-Forward Neural Networks: Processes the output of the attention mechanism.
   c. Residual Connections and Layer Normalization: Enhances training stability and gradient flow.
3. **Output Layer:** Produces the final predictions for classification tasks or generates context-aware representations for downstream tasks.

DistilBERT is trained using a process called **knowledge distillation** (see e.g. Gou et al., 2021) where a smaller model (the student) learns to mimic the behavior of a larger, pre-trained model (the teacher). The distillation process involves:

1. Teacher Model: The larger, pre-trained BERT model.
2. Student Model: The smaller DistilBERT model.
3. Loss Function: Combines the traditional supervised loss (e.g., cross-entropy) with a distillation loss, which measures the difference between the student and teacher model outputs.

The goal of distillation is to transfer the knowledge from the teacher model to the student model, allowing the smaller model to achieve similar performance.

Results of training/fine-tuning with DistilBERT

Training and Evaluation Setup:

- Number of epochs : 10
- Batch Size (Training and Evaluation) : 32
- Weight decay : 0.01

Performance Metrics - (trained on HASOC data - tested on HASOC data)

- **Accuracy :** 0.8596
- **Precision :** 0.8468
- **Recall   :** 0.8559
- **F1 Score :** 0.8096

Performance Metrics for 4 labels - (trained on HASOC data + our data - tested on our data - 80:20 split)

- **Accuracy :** 0.7241
- **Precision :** 0.7722
- **Recall   :** 0.7213

Social Media Governance: Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India

25

- **F1 Score :**   0.6436

Performance Metrics for 2 labels - (trained on HASOC data + our data - tested on our data - 80:20 split)

- **Accuracy :**   0.7759
- **Precision :**   0.7616
- **Recall   :**   0.8011
- **F1 Score :**   0.7471

7.4.   Some preliminary conclusions from our training experiments

Our studies with three LLMs all arrive at similar conclusions. The HASOC training set provides reasonable training for the held-out HASOC test set: our results here tally with the results obtained in the original HASOC work, and provide a good benchmark for our fine-tuning exercise. Fine-tuning with the small amount of data we have currently gathered doesn't allow us to reach this benchmark: in all cases, the fine-tuned model performs worse than the benchmark. This is to be expected, given we are fine-tuning on a language sample and annotation scheme that are somewhat different from the HASOC dataset (as noted in Sections 7.1 and 7.2), and also much smaller. But F1 scores for the fine-tuned models are promising, given these considerations—in particular the scores of 0.76 and 0.85 for the RoBERTa model. These give us some confidence that our method is capturing useful intuitions from our small 'crowd' of annotators.

8.   Our continuous annotation study of Tweets

In Sections 6 and 7, we discussed our discrete annotation study of Tweets, which we envisage using to make categorical content moderation actions: 'remove', 'downrank', 'leave untouched' and 'uprank'. As discussed in Section 3, if an item is to be downranked or upranked, we need additional information to decide how much down- or up-ranking should occur. This information comes from a second annotation process, that places Tweets on a *continuous* scale of 'hatefulness'. In this section, we discuss our pilot for this second process.

Recall from Section 3 that our continuous valuations of Tweets are created from a dataset of annotator judgements about *pairs* of Tweets, identifying for each pair which is 'more hateful'. We then use the Bradley-Terry model to convert these pair judgements into valuations for individual Tweets. In Section 8.1 we describe the annotation interface for pair judgements. In Section 8.2 we describe how we picked pairs of Tweets: we used a mixture of 'dense' and 'sparse' schemes for allocating pairs of Tweets to annotators. We describe two different Bradley-Terry models for creating continuous valuations of Tweets: the 'traditional' model in Section 8.3, and a more recent Bayesian variant in Section 8.4. We present the results of these models in Section 8.5, and a brief discussion in Section 8.6.

8.1. The annotation interface for pairwise judgements

We have created a separate *"Tweet Compare Annotation Form"* webpage on our existing server which is dedicated to full annotations. Similar to our 2023 platform, here a user has to annotate all the pairwise comparisons and cannot skip any of them. This can be found at https://annotate.infomaticae.com/.

Tweet Compare Annotation Form

25.
Tweet 1 : President of @AITC4Goa Shri @KandolkarKiran denies all fake news regarding disbandment of @AITC4Goa committee. These are desperate tricks by those who fear a "new dawn" in Goa says @KandolkarKiran. @ANewDawnForGoa #goencherajkaran #GoaElections2022 https://t.co/JuL9mXtrTX

Tweet 2 : #PakdaGayaModi 's choice for Rs 58000 cr #RafaleDeal was AA. Look at the bankruptcy of his entire group. #HAL unceremoniously dumped. Nothing can save the #ChowkidarChorHai . It is a matter of time. #Modi is a corrupt man. And add to it Birla-Sahara diaries, #GSPCScam etc. https://t.co/pvOJ9YQClq

Which one is more hateful?
○ Tweet 1
○ Tweet 2

104.
Tweet 1 : 'Uttarakhand polls contest between Harish Rawat and BJP's misrule' @INCUttarakhand @harishrawatcmuk #UttarakhandElections2022 #HarishRawat #Congress https://t.co/CulNE9Alxh

Tweet 2 : #AAP deep fake video getting exposed and Niti Gadkari exposing relationship with #BJP Will hurt them in #PunjabElections2022 as it's happening only next Sunday. To benefit of BJP they are losing other state. @INCPunjab should expose them. @Morewithshashi @partha2019LS

Which one is more hateful?
○ Tweet 1
○ Tweet 2

Figure 12. 'Tweet Compare' Annotation Form

8.2. Selection of pairs of Tweets to be annotated: 'dense' and 'sparse' datasets

We first created a set of 'Tweet pairs' for annotation, drawing from the set of over 1000 Tweets that received 'discrete' annotations. Since our primary focus is on deciding on operations for Tweets in the 'downrank' category, we selected with a bias towards Tweets identified as 'downrank' in the discrete annotation task.

We generated two datasets of pairs: a 'dense' set, featuring all possible pairings of a (small) set of Tweets, and a 'sparse' set, comprising pairs drawn randomly from the full set of Tweets. The dense set served to ensure a common reference point for all annotators—and also as a dataset to explore the Bradley-Terry model in its own right.

For the dense dataset, we first selected a set of 10 Tweets based on their assigned labels from annotators. Out of these, 6 Tweets were labeled either 'remove' or 'downrank' by the majority of annotators; 3 were labeled 'neutral', and 1 was labeled 'uprank'. This provides a bias towards the 'downrank' and 'remove' categories, while ensuring some representation from all categories. Once we had these 10 Tweets, we generated all possible pairs, resulting in 45 pairs (10 choose 2). For

the sparse dataset, we selected an additional 245 pairs from the remaining Tweets, choosing randomly, with the constraint that each Tweet appeared in at least one pair.

Each annotator was given a set of 290 Tweet pairs, comprising the 45 dense pairs, plus the 245 sparse pairs (the same pairs for every annotator). These pairs were interleaved, and presented to annotators in a random order. There were 25 annotators, as described in Section 4.2.

8.3.   The Bradley-Terry method for analysing pairwise judgements

The Bradley-Terry model is a probabilistic model used to predict the outcome of pairwise comparisons between items, based on their latent 'ability' or 'strength' (see Firth, 2005 for an introduction). It assumes that the probability of one item being preferred over another depends on the relative strengths of the two items.

Bradley-Terry Model Formulation

Let θ_i and θ_j represent the strengths of items *i* and *j*, respectively. The probability that item *i* is preferred over item *j* in a pairwise comparison is given by:

$$P(i \text{ is preferred over } j) = \frac{\theta_i}{\theta_i + \theta_j}$$

Here, the parameters θ_i and θ_j are non-negative values, often interpreted as the 'quality' or 'strength' of the respective items. The model assumes that the higher the strength of an item, the more likely it will be preferred over other items in pairwise comparisons.

Estimating θ_i and θ_j

To estimate the strengths θ_i for all items, the Bradley-Terry model uses the maximum likelihood estimation (MLE). Given a set of pairwise comparisons, where n_{ij} is the number of times item *i* is preferred over item *j*, the likelihood function can be written as:

$$L(\theta_1, \theta_2, \ldots, \theta_N) = \prod_{i,j} \left(\frac{\theta_i}{\theta_i + \theta_j} \right)^{n_{ij}}$$

Taking the logarithm of the likelihood function (log-likelihood) for simplification:

$$\log L(\theta_1, \theta_2, \ldots, \theta_N) = \sum_{i,j} n_{ij} \log \left(\frac{\theta_i}{\theta_i + \theta_j} \right)$$

This log-likelihood function is then maximized with respect to θ_i to estimate the item strengths. Various numerical optimization methods, such as iterative algorithms, can be used to solve this maximization problem.

Social Media Governance: Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India

28

Deriving a Continuous Scale from Pairwise Comparisons

The estimated θ_i's from the Bradley-Terry model provide a relative measure of strength or quality. However, these values are on a multiplicative scale. To interpret them on a continuous scale (such as between -2 and 1 – with respect to the labels defined in our Tweet dataset), a transformation can be applied. One common approach is to take the log of θ_i:

$$\text{score}_i = \log(\theta_i)$$

This transformation maps the strengths θ_i onto a continuous scale. To further normalize these scores to fit within a specific range (e.g., [-2, 1]), we apply a linear transformation:

$$\text{score}'_i = a \cdot \log(\theta_i) + b$$

where *a* and *b* are constants chosen to scale and shift the scores as needed.

Therefore, the Bradley-Terry model provides a probabilistic framework for pairwise comparisons, and the estimated strengths can be converted into a continuous scale through logarithmic transformation, followed by scaling and shifting if necessary

8.4. A Bayesian version of the Bradley-Terry model

In a Bayesian approach to the Bradley-Terry model, instead of estimating point estimates for the item strengths θ_i, we treat them as random variables with prior distributions (see e.g. Caron and Doucet, 2012). This approach allows us to incorporate prior information (such as discrete scores) and quantify uncertainty in the estimates of item strengths.

Likelihood function

The likelihood function for pairwise comparisons remains the same as in the classical Bradley-Terry model. For each comparison between items *i* and *j*, the probability that item *i* is preferred over item *j* is:

$$P(i \text{ is preferred over } j) = \frac{\theta_i}{\theta_i + \theta_j}$$

If y_{ij} represents the outcome of a comparison between items *i* and *j* (with y_{ij} = 1 if i is preferred and y_{ij} = 0 if *j* is preferred), the likelihood for all pairwise comparisons can be written as:

$$P(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i,j} \left(\frac{\theta_i}{\theta_i + \theta_j} \right)^{y_{ij}} \left(\frac{\theta_j}{\theta_i + \theta_j} \right)^{1-y_{ij}}$$

Prior distributions

In the Bayesian framework, we must assign a prior distribution to each item's strength θ_i. A common choice is to use a log-normal prior or a Gamma prior, since the strengths θ_i are positive. If we know some discrete scores of the items, we can choose a prior that reflects this prior knowledge.

Suppose the discrete score for item i is denoted by s_i, and we assume the relationship:

$$\log(\theta_i) \sim \mathcal{N}(s_i, \sigma^2)$$

This implies that the logarithm of the item strength is normally distributed around the discrete score s_i with variance σ^2 . The prior for θ_i becomes:

$$P(\theta_i|s_i) = \frac{1}{\theta_i\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log(\theta_i) - s_i)^2}{2\sigma^2} \right)$$

Incorporating Discrete Scores into the Continuous Scale

We opted to use the discrete scores s_i, assigned in the discrete annotation exercise, as our priors, to guide the Bayesian model towards estimates that are consistent with this earlier exercise. The resulting continuous scores are a combination of the prior belief (from s_i) and the observed pairwise comparisons. This approach provides a more nuanced and flexible way to derive continuous scores than the classical Bradley-Terry model, particularly when prior information about item strengths is available.

In summary, the Bayesian approach to the Bradley-Terry model modifies the classical formulation by treating the item strengths θ_i as random variables with priors that reflect prior knowledge (e.g., discrete scores). The continuous score is then derived from the posterior distribution over θ_i, often using the posterior mean or median, and can be transformed as needed for interpretability.

8.5. Results from the Bradley-Terry models

There is no obvious benchmark for the continuous valuations for Tweets produced by the Bradley-Terry models. The whole project of crowdsourcing is founded on the idea that no single judge can provide an authoritative valuation. Ultimately, it would be useful to consult 'domain experts', to see how well their judgements align with the results of our Bradley-Terry models. And we intend to do this in subsequent work. But for now, we present preliminary evaluations simply by comparing the Bradley-Terry valuations of Tweet 'hatefulness' with the discrete labels assigned by

expect there will be some consistency between the continuous and discrete measures derived from
the two exercises. To assess this visually and quantitatively, we place the four discrete categories
(somewhat arbitrarily) at four points on a continuum: -2 for 'remove', -1 for 'downrank', 0 for 'neutral',
and 1 for 'uprank'. We can then plot discrete and continuous measures against one another,
and—as a very preliminary measure—compute correlation coefficients, to assess consistency. (The
correlation measure should be interpreted with great care for the Bayesian model, given that it uses
discrete classes as its prior—but there are certain effects that are still worth noting.)

We use Spearman's rank correlation coefficient (ρ) to measure correlation. This is a non-parametric
measure used to assess the strength and direction of a monotonic relationship between two
variables. Unlike Pearson correlation, which measures linear relationships, Spearman correlation is
based on the ranks of the data rather than their raw values, making it useful when the data does not
meet normality assumptions or has outliers. The coefficient ranges from -1 to +1, where +1 indicates
a perfect positive monotonic relationship, -1 indicates a perfect negative monotonic relationship, and
0 signifies no monotonic relationship.

8.5.1.   Results from the classical Bradley-Terry model

Verifying our implementation

As a quick reality check, we used some synthetic data to validate our Bradley-Terry implementation.
We created 10 items, ranked 1-10. We then created all possible pairings of these items (10 choose
2, resulting in 45 pairs) and assign 'choices' in line with item rankings. ankings. We then used these
'choices' as input into our implementation to produce Bradley-Terry rankings, which we compare
with the original rankings for accuracy. As depicted in Figure , the Bradley-Terry rankings align with
the original rankings, confirming that our implementation functions correctly.

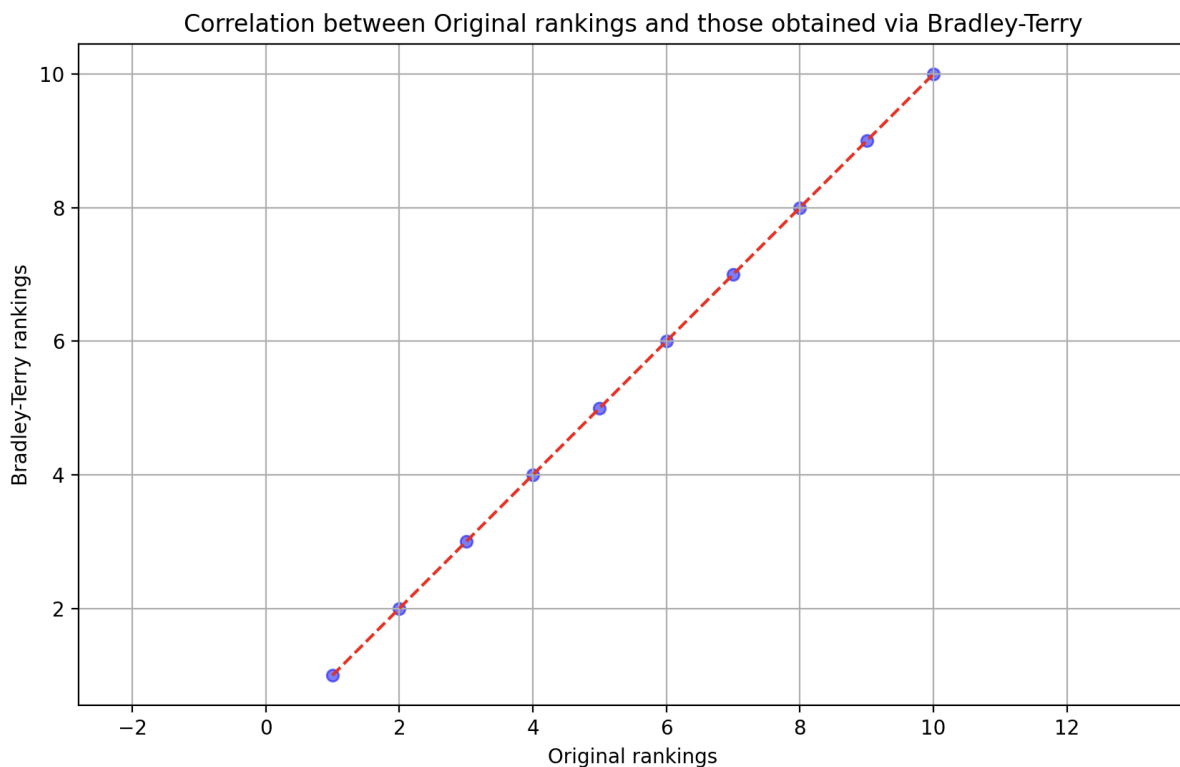Correlation between Original rankings and those obtained via Bradley-Terry

Figure 13. Validation of the classical Bradley-Terry model with synthetic data

Correlation results for the 'sparse' dataset of pair judgements

As shown in Figure 14, the annotations for the 290 Tweet pairs created 'sparsely' from the full dataset yielded a very low correlation coefficient (0.02). Most Tweets received only a single score on the continuous scale, as they appeared in only one or a few comparisons, making it difficult to assign them an accurate score.
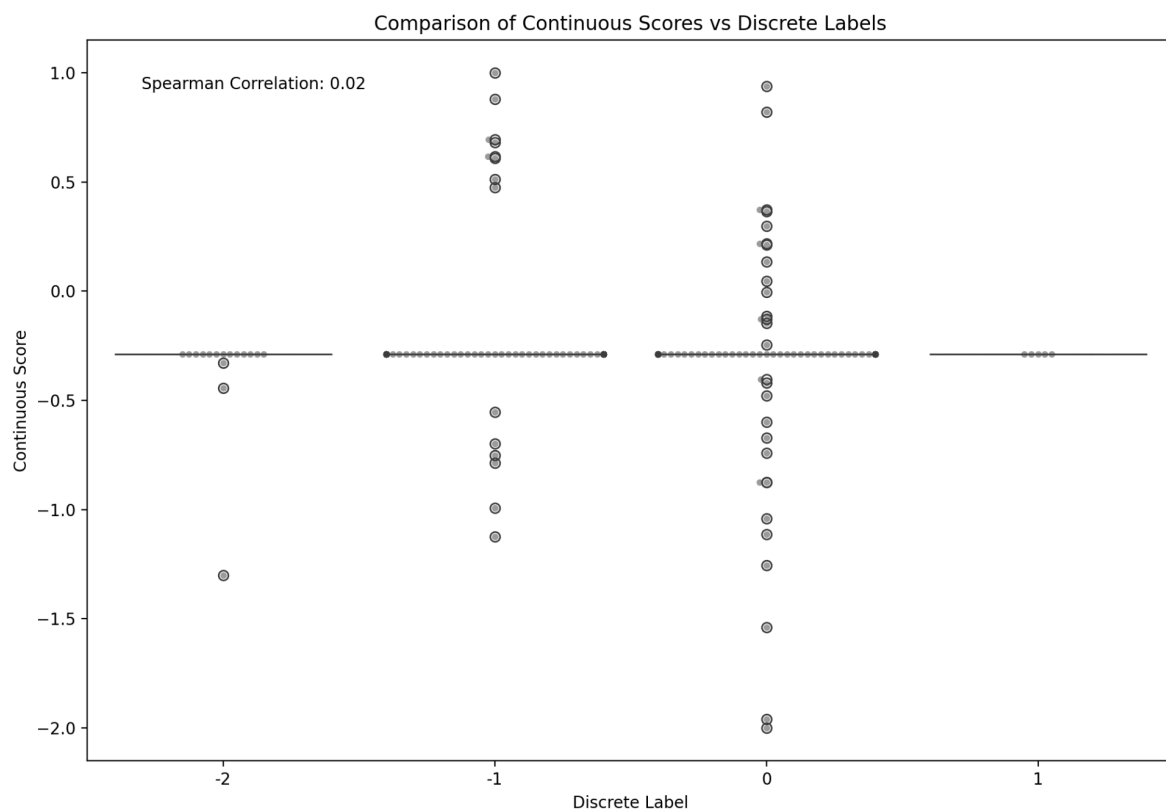
Figure 14. Comparison of classical Bradley-Terry scores with discrete labels, for the 'sparse' dataset

Correlation results for the 'dense' dataset of pair judgements

Correlation results for the 'dense' set of 45 Tweet pairs showing all combinations of 10 selected Tweets are shown in Figure 15. For this dataset, the correlation coefficient is significantly higher (0.64). This is to be expected, because a great deal more information is provided about the ranking of these items in the dense dataset. Correlation is still not perfect, of course, because there are many annotators, making different decisions.
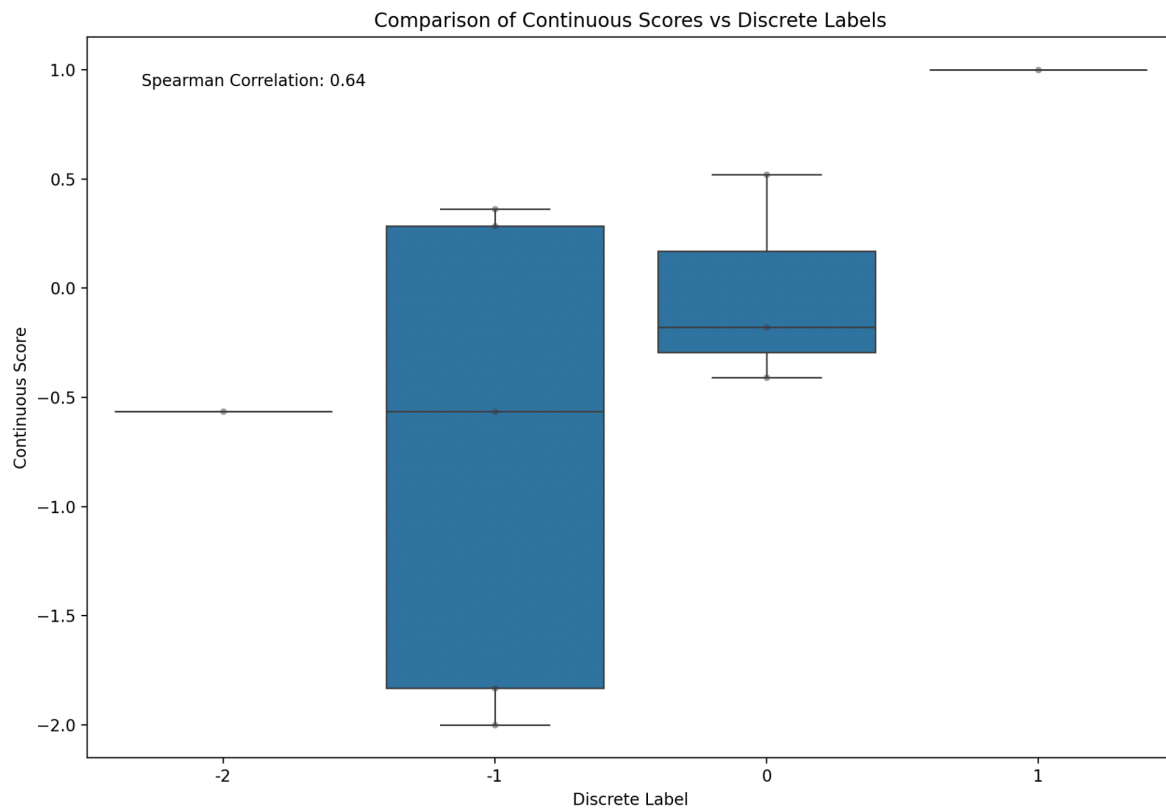
Figure 15. Comparison of classical Bradley-Terry scores with discrete labels, for the 'dense' dataset

8.5.2. Results from the Bayesian Bradley-Terry model

Correlation results for the complete dataset of pair judgements

Correlation results for the Bayesian version of the Bradley-Terry model, on the complete set of pair judgements, are shown in Figure 16. Unlike the standard Bradley-Terry method, the Bayesian approach yields a much higher correlation (0.92)—but as noted, this is to be expected, because of the way priors are computed in the Bayesian model. Because of priors, Tweets that appear in fewer pairwise comparisons—the great majority of Tweets—naturally align with the given prior information.
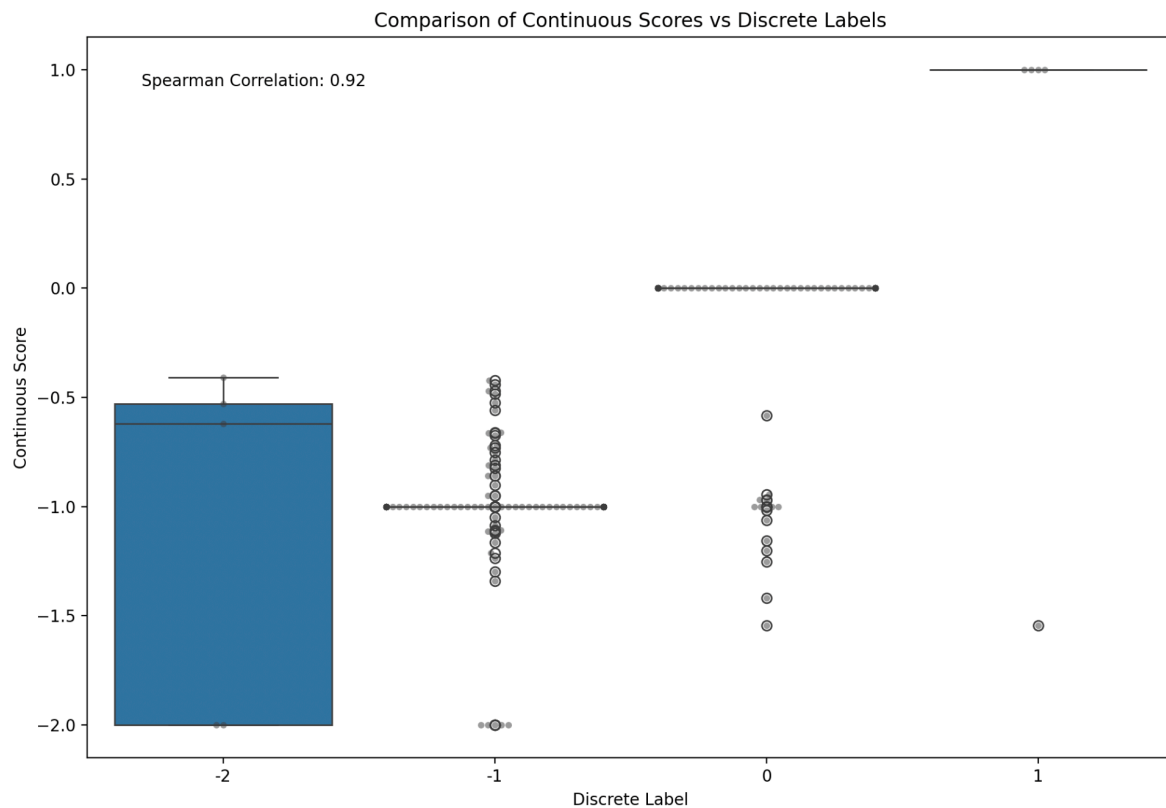
Figure 16. Comparison of Bayesian Bradley-Terry scores with discrete labels, for the complete dataset of pair judgements

Correlation results for the 'dense' dataset of pair judgements

Correlation results for the 45 'dense' pairs of Tweets for the Bayesian Bradley-Terry model are shown in Figure 17. Here, the correlation coefficient is higher than in the standard Bradley-Terry model, as it incorporates prior information from the discrete labels. However, unlike the results for all Tweets, it is not entirely skewed toward the discrete labels. This demonstrates that this round of annotations validates the results from the first phase and helps improve downranking by offering a more refined continuous score.

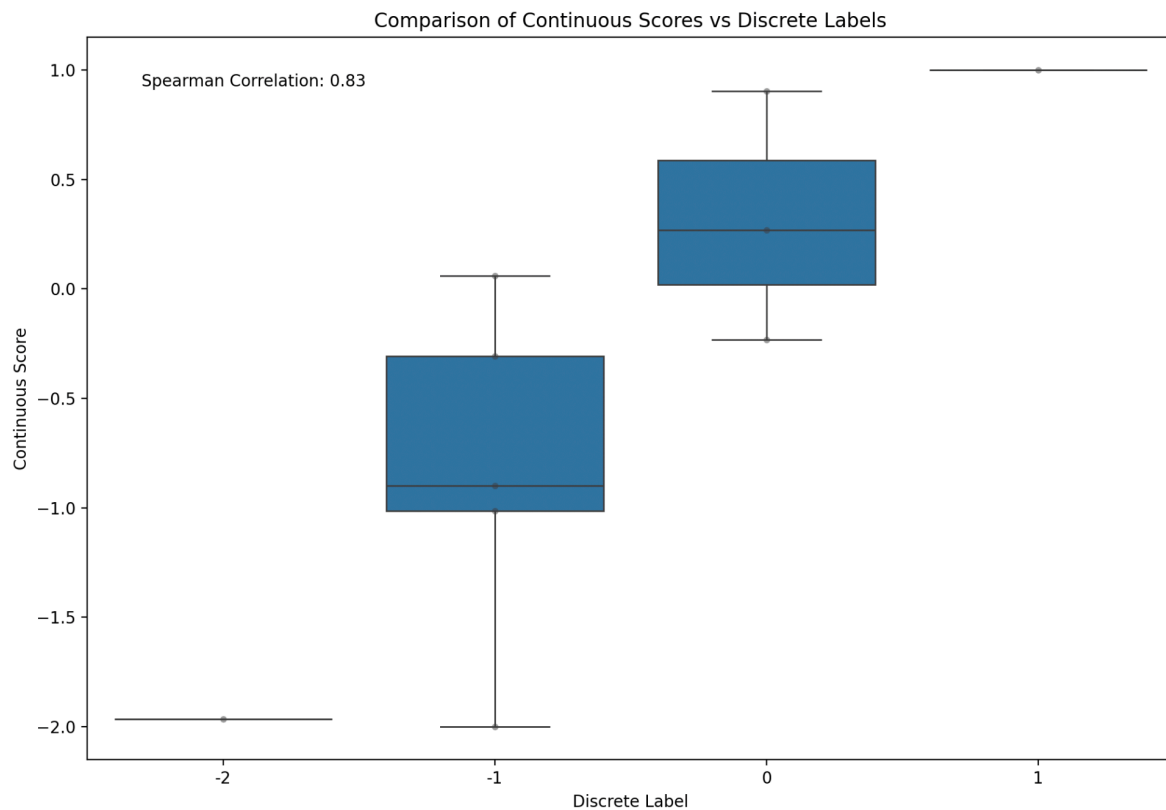**Social Media Governance:** Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India

34

Figure 17. Comparison of Bayesian Bradley-Terry scores with discrete labels, for the 'dense' pair judgements

8.6. Discussion

As already noted, there are some challenges in assessing our continuous annotation exercise, because there's no obvious 'ground truth' set of valuations to measure against. In due course we plan to ask some 'local experts' in hate speech for their continuous rankings. But there are known problems asking informants to directly assign values to items on a continuous scale (see e.g. Goffin and Olson, 2011). This is what motivates us to ask annotators for judgements about pairs in the first place. There are also problems with identifying people as 'experts', in this area where opinions play an important role in judgements. As a preliminary evaluation, we have reported how consistent our Bradley-Terry models are with the discrete judgements made by annotators in our first annotation study. We can draw some cautious conclusions from our findings.

One finding is that 'denser' datasets of judgement pairs seem to allow the model to operate more effectively than 'sparser' datasets. This is not surprising: they simply provide more information. But the poor results on our very sparse dataset with the classical Bradley-Terry model) indicate that at least this model needs a certain minimum level of density before it will give meaningful valuations.

The Bayesian version of the Bradley-Terry model may help to compensate for data sparsity. At present it's hard to tell, because our evaluation method is tainted by the way we choose priors for the model. When we have asked experts for valuations of Tweets, we will have two independent valuations: we can use one to define priors for the Bayesian model, and the other as a yardstick for

correlations. At that point, we will have a better sense of whether the Bayesian method provides a practical improvement over the classical method in this domain.

We might also address data sparsity by picking different pairs of items for each annotator in the 'sparse' dataset. This would certainly cover more pairs, at the expense of losing information about disagreements for a specific pairs. We may also need to explore other assignment schemes, in between our current dense and sparse schemes.

9. Our discrete annotation study of memes

9.1. The annotation interface for the discrete study of memes

We have created a separate *"Meme Annotation Form"* webpage on our existing server. This interface is configured for 'dense': the user has to annotate all the memes and cannot skip any of them. We have also extended our front-end and back-end software, to handle images, rather than texts.

The meme annotation interface can be found at https://annotate.infomaticae.com/. Screenshots of different windows, showing the annotation form and a summary page, are shown in Figures 18 and 19.

Figure 18. Meme Annotation Form

Social Media Governance: Crowdsourcing annotations for harmful content classifiers: an update from GPAI's pilot project on political hate speech in India
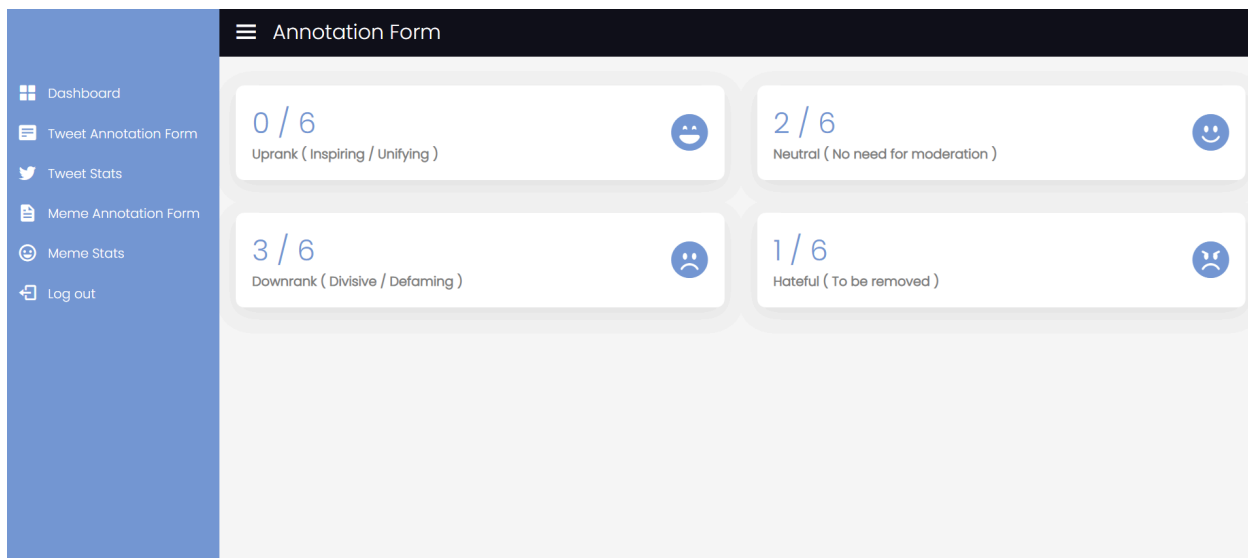
36

Figure 19. Meme stats of a user by label class

9.2. An analysis of data in the discrete study of memes

This extended part of our training dataset contains 355 memes (as described in Section 5.2), and was annotated by a total of 20 annotators (as described in Section 5.2). In this section, we report some preliminary findings. As for our discrete Tweet annotation study, our first analysis focusses on the disagreement that was found between annotators.

For analysis purposes, we first separate out the memes that were 'reported' (i.e., flagged as unclassifiable) by a majority of annotators. There were 9 of these. The remaining memes were further analyzed into three discrete classes, based on the amount of agreement between the annotators:

- ○ Unanimous Agreement (27 memes)
- ○ Disagreement by a degree of 1 (19 memes)
- ○ Disagreement by a degree of 2 (93 memes)
- ○ Disagreement by a degree of 3 (207 memes).

These results are depicted in Figure 20.

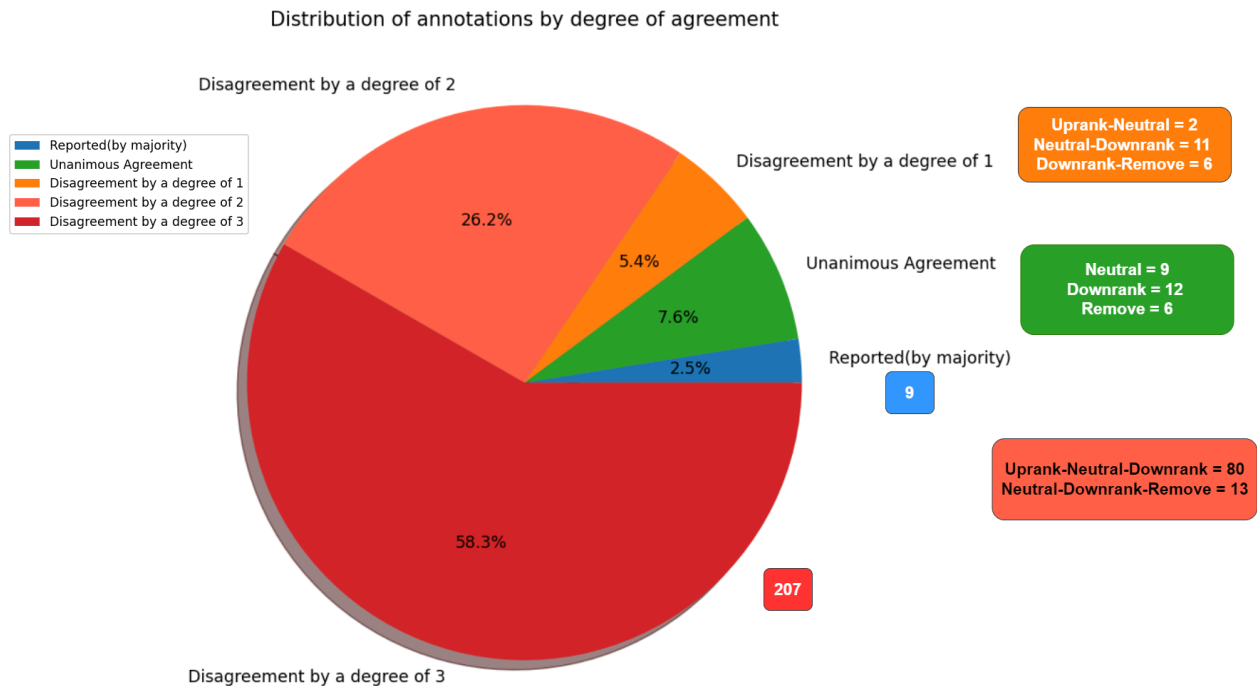Distribution of annotations by degree of agreement

Figure 20. Distribution of annotations by degree of agreement

Figure 20 also displays the distribution of memes across categories of Unanimous Agreement, Disagreement by a degree of 1, and Disagreement by a degree of 2, based on their corresponding class labels.

As we can observe, there are varying degrees of disagreement across the dataset—including a major portion of memes for which there is significant disagreement. This result is not unexpected across a team of 20 annotators, but it is useful in providing preliminary evidence that the amount of disagreement will vary significantly over content items.

Finally, we report an entropy analysis of the annotations for this dataset. Figure 21 shows a histogram of entropy ranges for the dataset. The memes with most disagreement shown in the previous Figure have entropies from the highest entropy ranges in the Figure below.
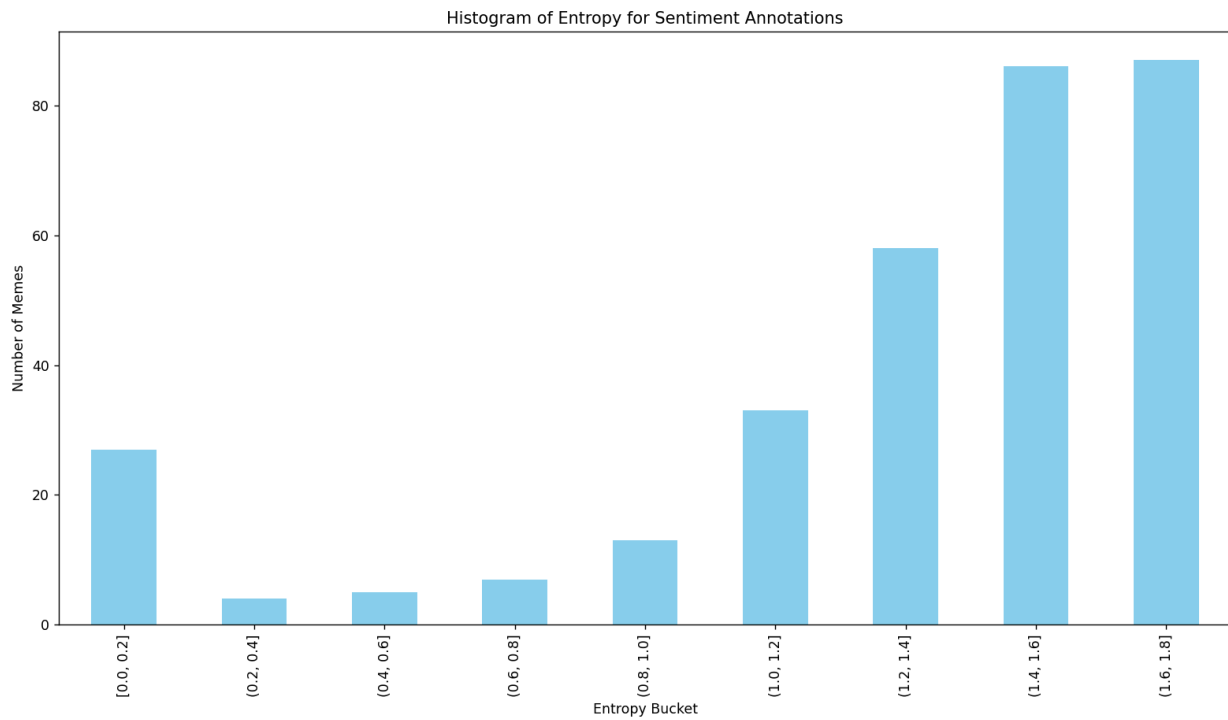
Histogram of Entropy for Sentiment Annotations

Figure 21. Histogram of Entropy for the discrete meme annotation task

10.   Summary and future work

In this report, we have described our continuing investigation of the idea that citizens can be entrusted with the task of training harmful content classifiers for social media platforms, in ways that increase the transparency, accountability, and efficiency of classifiers, and associated content moderation processes. We have explored this idea for one particular kind of harmful content—political hate speech—in one particular part of the world—India. The experiments we have reported here are pilot studies in this domain, rather than full-scale evaluations. Their purpose is to create suitable annotation tools and protocols, and to identify suitable methods for processing the annotations which are produced by these methods.

The purpose of a pilot study is to determine whether further studies are warranted. For memes, there are still some further pilot studies to conduct before we can judge this. But for Tweets, we believe the pilot studies we have described here clearly indicate that a larger annotation study is warranted. Many of the questions which remain unresolved in the current report would be addressed in a larger study, gathering more data from citizens. More data would improve the accuracy of Tweet classifiers, trained using the methods described in Section 7. More data would also improve the prospect of training a regression model to downrank Tweets, informed by the methods described in Section 8. We are certainly ready to scale up the relevant annotation exercises: the methods we have described here have been designed to scale. We believe a scaled-up version of the Tweet annotation project piloted here would be a valuable contribution to the current debate about the best methods to implement content moderation in social media.

References

Adel, Hadeer & Dahou, Abdelghani & Mabrouk, Alhassan & Elsayed Abd Elaziz, Mohamed & Kayed, Mohammed & El-henawy, Ibrahim & Alshathri, Samah & Ali, Abdelmgeid. (2022). Improving Crisis Events Detection Using DistilBERT with Hunger Games Search Algorithm. Mathematics. 10. 447. 10.3390/math10030447.

Caron, F., & Doucet, A. (2012). Efficient Bayesian inference for generalized Bradley–Terry models. Journal of Computational and Graphical Statistics, 21(1), 174-196.

Chakravarthi, B. R., Anand Kumar M, McCrae, J. P., Premjith, B., Soman, K. P., & Mandl, T. (2020). Overview of the track on HASOC-Offensive Language Identification-DravidianCodeMix. In FIRE (Working notes) (pp. 112-120).

Das, Mithun, et al. (2024). Low-Resource Counterspeech Generation for Indic Languages: The Case of Bengali and Hindi." arXiv preprint arXiv:2402.07262.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics.

Dowlagar, S., & Mamidi, R. (2021). A survey of recent neural network models on
code-mixed Indian hate speech data. In Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation (pp. 67-74).

El Asam, A., & Samara, M. (2016). Cyberbullying and the law: A review of psychological and legal challenges. Computers in Human Behavior, 65, 127-141.

Firth, D. (2005). Bradley-Terry models in R. Journal of Statistical software, 12, 1-12.

Goffin, R. D., & Olson, J. M. (2011). Is it all relative? Comparative judgments and the possible improvement of self-ratings and ratings of others. Perspectives on Psychological Science, 6(1), 48-60.

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. International Journal of Computer Vision, 129(6), 1789-1819.

GPAI (2023). Crowdsourcing the curation of the training set for harmful content classifiers
used in social media: A pilot study on political hate speech in India, Report, November 2023, Global Partnership on AI.

Huang, Zihan, et al. (2021). Context-aware legal citation recommendation using deep learning. Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law.

Kumar, R., Reganti, A. N., Bhatia, A., & Maheshwari, T. (2018). Aggression-annotated corpus of hindi-english code-mixed data. arXiv preprint arXiv:1803.09402

Lan, Z. ALBERT: A lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942(2019).

Liu, Yinhan (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., & Patel, A. (2019). Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages. In Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation (pp. 14-17).

Mandl, T., Modha, S., Kumar M, A., & Chakravarthi, B. R. (2020). Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German. In Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation (pp. 29-32).

Mandl, T., Modha, S., Shahi, G. K., Madhu, H., Satapara, S., Majumder, P., ... & Jaiswal, A. K. (2021). Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages. arXiv preprint arXiv:2112.09301.

Meta (2024). Hate Speech. Meta Transparency Center. https://transparency.meta.com/en-gb/policies/community-standards/hate-speech/ (Accessed October 2024)

Mirchandani, M. (2018). Digital hatred, real violence: Majoritarian radicalisation and social media in India. ORF Occasional Paper, 167, 1-30.

Modha, Sandip, et al. "Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech." Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation. 2021.

Murphy, M. R. (2019). Context, content, intent: Social media's role in true threat prosecutions. U. Pa. L. Rev., 168, 733.

Paz, M. A., Montero-Díaz, J., & Moreno-Delgado, A. (2020). Hate speech: A systematized review. Sage Open, 10(4), 2158244020973022.

Romim, Nauros, et al (2021). Hate speech detection in the Bengali language: A dataset and its baseline evaluation. Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020. Springer Singapore.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Saroj, A., & Pal, S. (2020). An Indian language social media collection for hate and offensive speech. In Proceedings of the Workshop on Resources and Techniques for User and Author

Profiling in Abusive Language (pp. 2-8).

Uma, A. N., Fornaciari, T., Hovy, D., Paun, S., Plank, B., & Poesio, M. (2021). Learning from disagreement: A survey. Journal of Artificial Intelligence Research, 72, 1385-1470.

Voorhees and Harman (2005). *TREC. Experiment and Evaluation in Information Retrieval.* MIT Press.

Zhou, Faguo & Wang, Chao & Wang, Jipeng. (2022). Named Entity Recognition of Ancient Poems Based on Albert-BiLSTM-MHA-CRF Model. Wireless Communications and Mobile Computing. 2022. 1-11. 10.1155/2022/6507719.