Proceedings of the Twentieth International AAAI Conference on Web and Social Media (ICWSM 2026)

LaMSUM: Amplifying Voices Against Harassment through LLM Guided Extractive Summarization of User Incident Reports

Garima Chhikara[1,2], **Anurag Sharma**[3], **V. Gurucharan**[4]
Kripabandhu Ghosh[5], **Abhijnan Chakraborty**[3]
[1] Indian Institute of Technology Delhi, India
[2] Delhi Technological University, India
[3] Indian Institute of Technology Kharagpur, India
[4] Collaborative Dynamics, Texas, USA
[5] Indian Institute of Science Education and Research Kolkata, India

Abstract

Citizen reporting platforms help the public and authorities stay informed about sexual harassment incidents. However, the high volume of data shared on these platforms makes reviewing each individual case challenging. Therefore, a summarization algorithm capable of processing and understanding various code-mixed languages is essential. In recent years, Large Language Models (LLMs) have shown exceptional performance in NLP tasks, including summarization. LLMs inherently produce abstractive summaries by paraphrasing the original text, while the generation of extractive summaries – selecting specific subsets from the original text – through LLMs remains largely unexplored. Moreover, LLMs have a limited context window size, restricting the amount of data that can be processed at once. We tackle these challenges by introducing `LaMSUM`, a novel multi-level framework combining summarization with different voting methods to generate extractive summaries for *large* collections of incident reports using LLMs. Extensive evaluation using four popular LLMs (Llama, Mistral, Claude and GPT-4o) demonstrates that `LaMSUM` outperforms state-of-the-art extractive summarization methods. Overall, this work represents one of the first attempts to achieve extractive summarization through LLMs, and is likely to support stakeholders by offering a comprehensive overview and enabling them to develop effective policies to minimize incidents of unwarranted harassment.
Warning: This paper contains content that may be disturbing or upsetting.

1 Introduction

In recent decades, the widespread availability of the internet has provided seamless access to online platforms to millions of people. Governments worldwide are increasingly utilizing these platforms to gather information directly from citizens – referred to as *Citizen Reporting* (Kopackova and Libalova 2019). By leveraging tools such as mobile applications, web-based portals, and social media integrations, citizen reporting platforms establish a direct and efficient communication link between individuals and the relevant authorities, enabling faster issue resolution and facilitating active public participation in community improvement. Beyond immediate problem solving, real-time data gathered through these platforms contributes valuable information for urban planning and proactive measures, paving the way for more efficient and adaptive communities. Citizen reporting typically addresses topics such as community issues, environmental challenges, crime prevention, public health, and disaster response (Shin et al. 2024).

A special category of citizen reporting platforms, such as *Safe City* (https://webapp.safecity.in), *SHe-Box* (https://shebox.wcd.gov.in), and *JDoe* (https://jdoe.io), allow people to post incidents of sexual harassment, domestic abuse, violence and assault. Table 1 showcases some example incidents shared by users on one such platform. While such horrific incidents cannot be entirely avoided through reporting alone, these *incident reporting platforms* can play a crucial role in preventing certain cases of sexual assault. By enabling users to analyze reported incidents, assess the safety of specific locations, and make informed decisions when traveling to potential hotspots, these platforms contribute to enhanced personal safety and awareness. Similarly, the local authorities can also benefit from these platforms to assess emerging cases, identify the underlying factors and determine proactive measures for effective resolution. However, the challenge for the authorities is to navigate the high volume of information in such platforms. Manually reviewing all posts is often impractical, *necessitating a summarization algorithm that can identify and select posts that are diverse as well as representative of the original data*. Additionally, incident reporting platforms often feature a curated selection of posts on their homepage to showcase their core purpose, mission, and services. *This deliberate selection also acts as a form of summarization*.

Summarization algorithms are of two types: 'extractive' and 'abstractive'. In *extractive summarization*, the algorithm selects a subset representative of the original text (Xu et al. 2020; Zhong et al. 2020; Zhang, Liu, and Zhang 2022; Dash et al. 2019; Zhang, Liu, and Zhang 2023a). In contrast, *abstractive summarization* algorithms generate summaries that capture the essence of the original text, often paraphrasing the content (Pu, Gao, and Wan 2023). For incident reporting platforms, extractive summarization is more suitable, as the goal is not to paraphrase the posts but to select a few that accurately capture a snapshot of the original content. *When summarizing such sensitive posts, preserving the user's ex-*

Category	Post
Robbery	This incident took place in the evening.Two bikers came on a bike and snatched Rs.17000 from an old lady at gun point.
Stalking	I was stalked by a guy who followed me for days and also he sent me letters on my doorstep saying he was madly in love with me and he is obsessed with my body.
Sexual Invites	I was walking on footpath to a place nearby to meet my friends. A man was driving his car on parallel road and was continuously passing vulgar comments. I ignored but 15 min later he stopped his car and said *"chalri h ky, paise h mere pas"* (I have money, you want to come with me?). I screamed and asked for help from people around me.
Mas*****tion in public	One day I reached my school prior to the school timings mistakenly. One of the drivers called me and my friend and started mas*****ting in front of us.
Ogling	I was going to my coaching by driving my scooty when suddenly some boys came on a bike. They started making cheap comments and teasing me. It was late and the area was secluded. I got scared as they started revolving their bike around me. I started driving in the direction of a crowded place that is when they left.
Showing Po**ography	A man in a car parked outside was watching po** when I saw him. He turned his device towards me and did offensive hand gestures inviting me.
Sexual Assault	When I was 7 years old, the shopkeeper removed my clothes and started touching me everywhere. He also tried to do it to another girl and failed.
Domestic Violence	My husband always doubts on my character and doesn't allow me to go outside alone, uses very vulgar language and beats me.

Table 1: Examples of harassment cases shared on an incident reporting platform. Proactive action by authorities and citizens can help prevent numerous such incidents. Providing stakeholders with a concise overview of incidents occurring in a specific area is crucial and this can be effectively achieved by utilizing summarization algorithms.

act words is essential, making extractive summarization particularly valuable in maintaining authenticity and context.

Several extractive summarization algorithms for user generated content have been proposed in the literature, primarily for text written in English (Bhattacharya et al. 2021; Kanwal and Rizzo 2022; Mukherjee et al. 2020; Jia et al. 2020). But there are several countries where English is not the primary language and users frequently communicate in code-mixed forms. For instance, India recognizes 22 official languages and users often post in Hinglish (a mix of Hindi and English). Such multilinguality limits the applicability of existing algorithms for extractive summarization of incident posts.

In recent years, Large Language Models (LLMs) have demonstrated very good performance across various tasks in multilingual and code-mixed settings (Ouyang et al. 2022; Brown et al. 2020; Tang et al. 2023a; Jin et al. 2024b). Plus, summaries generated by LLMs showcase high coherence and are overwhelmingly preferred by human evaluators over other baseline algorithms (Pu, Gao, and Wan 2023; Liu et al. 2024). These prior results motivated us to investigate the utility of LLMs for extractive summarization of large volumes of user generated posts. However, we encountered two significant limitations which hinder the immediate application of LLMs for extractive summarization:

1. As generative models, LLMs perform abstractive summarization by paraphrasing rather than selecting the most relevant sentences (as shown in Figure 1) (Worledge, Hashimoto, and Guestrin 2024).
2. Due to the finite size of the context window, LLMs cannot handle long texts in a single input, underscoring the need for a method that allows for processing long text (Jin et al. 2024a).

To overcome these limitations, in this paper, we present a novel framework `LaMSUM` (**La**rge Language **M**odel based Extractive **SUM**marization) that integrates LLM-generated summaries with voting algorithms borrowed from Social Choice Theory (Brandt et al. 2016). Our judicious application of voting algorithms with a multi-level summarization framework ensures that `LaMSUM` outperforms the state-of-the-art fine-tuned summarization models. In summary, in this work, we make the following contributions:

- We propose a novel framework `LaMSUM` which can produce extractive summaries from large (having $>$30K tokens) collection of user generated content. `LaMSUM` considers a multi-level summarization model that utilizes voting algorithms to combine LLM outputs to generate robust summaries.
- Extensive experiments with incident posts demonstrate that `LaMSUM` outperforms the state-of-the-art extractive summarization algorithms.

To our knowledge, this is the first work to implement extractive summarization of a large collection of user-generated texts using LLMs by combining summarization with voting algorithms. At the same time, we demonstrate the effectiveness of such algorithms to facilitate data-driven decision-making promoting safer communities by providing actionable insights into reported incidents. Code is available at https://github.com/garimachhikara128/LaMSUM

2 Background and Related Work

In this section, we review the relevant prior works that provide the foundation for our current research.

AI Solution through Citizen Reporting

Web and social media platforms receive posts on sensitive issues such as online harassment, hate speech, abusive behavior, violence etc. Abuse experienced by users leads to mental stress often forcing them to leave the platform (Sambasivan et al. 2019; Thomas et al. 2022; Kim et al. 2024). Several AI-powered solutions have been designed to address solutions to these critical issues. Machine learning based classi-

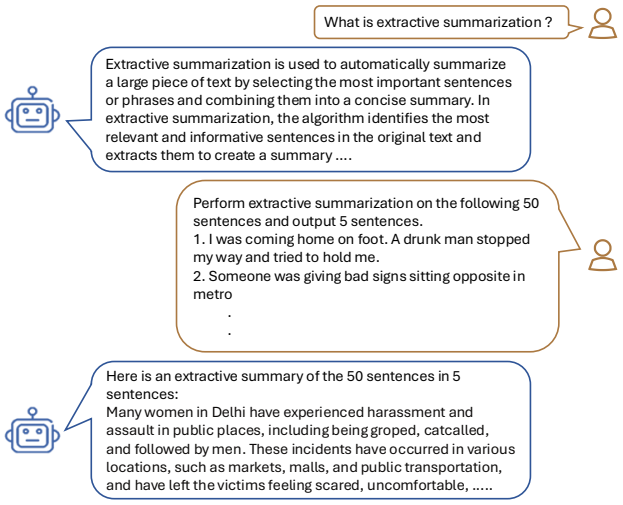

Figure 1: Current LLMs, by default, produce abstractive summaries. `Llama-3.3-70b`, despite specifically prompted for extractive summarization, generates abstractive summaries. This behavior underscores the need for a targeted approach to enable LLMs to effectively generate extractive summaries.

fiers and language models are utilized to detect the cases of sexual abuse, hate speech, offensive language, human trafficking and harassment cases (Sawhney et al. 2021; Hassan et al. 2020; Davidson et al. 2017; Singh, Bhattacharjee, and Chakraborty 2025; Upadhayay, Lodhia, and Behzadan 2021; Stoop et al. 2019; Ghosh Chowdhury et al. 2019). Modelling the cyberbullying behavior using social network and language-based features can improve the classifier performance (Ziems, Vigfusson, and Morstatter 2020; Olteanu et al. 2018). Development of a mobile computing based reporting tool empowers individuals with intellectual and developmental disabilities (I/DD) to self report abuse and share the incident with the intended group (Venkatasubramanian et al. 2021; Sultana et al. 2021). With the rise of public figures encouraging women to speak up, the number of non-anonymous self-reported assault stories has increased (ElSherief, Belding, and Nguyen 2017). Counterspeech is proven to be a viable alternative to blocking or suspending problematic messages or accounts, as it better aligns with the principles of free speech (Mathew et al. 2019). Conversational agents (CAs) have attracted significant interest as potential counselors due to their features, such as anonymity, which can help address many challenges associated with human-human interaction (Park and Lee 2021).

Large Language Models (LLMs) for Summarization

LLMs are now being extensively used for summarization (Brown et al. 2020; Tang et al. 2023a; Jin et al. 2024b). Multiple works have proposed few-shot learning frameworks for the abstractive summarization of news, documents, webpages, and generic texts (Zhang, Liu, and Zhang 2023b; Tang et al. 2023b; Yang et al. 2023; Bražinskas, Lapata, and Titov 2020; Laskar et al. 2023), but their primary focus remains on short documents that can fit in the LLM context window. Researchers have also observed that human evaluators are increasingly preferring LLM-generated summaries compared to other baselines (Zhang et al. 2024; Wu et al. 2024; Goyal, Li, and Durrett 2023; Zhang, Liu, and Zhang 2023c; Liu et al. 2024). Despite the advancements, recent studies have also uncovered factual inaccuracies and inconsistencies in LLM-generated summaries (Tang et al. 2024; Tam et al. 2023; Luo, Xie, and Ananiadou 2023; Laban et al. 2023).

Extractive Summarization through LLMs: The Current State

By default, LLMs produce abstractive summaries, meaning that the summary text is distinct from the input text, even when it is instructed to do otherwise. To illustrate this, we present a small example in Figure 1. An LLM, when prompted, could clearly explain extractive summarization, yet, when we instructed it to perform extractive summarization on a set of 50 sentences, it failed to do so and instead generated an abstractive summary. Prior to our current work, only two studies attempted to perform similar tasks. Zhang et al. attempted summarization of short news articles using GPT 3.5 (Zhang, Liu, and Zhang 2023b) , while Chang et al. attempted abstractive summarization for book-length documents (Chang et al. 2024a). However, both these approaches suffer from practical limitations such as lack of contextual dependencies in user generated text and the problem with positional bias.

To the best of our knowledge, ours is the first attempt to perform extractive summarization on a large collection of user generated texts through LLMs, while tackling the challenge of positional bias. We describe our proposal in detail in the next section.

3 `LaMSUM`: Generating Extractive Summaries through LLMs

In this section, we define the problem statement formally and introduce our novel summarization framework `LaMSUM` (**La**rge Language **M**odel based Extractive **SUM**marization) that leverages LLMs to summarize large user-generated text.

3.1 Task Formulation

Let $\mathcal{T} = \{t_1, t_2, \ldots t_N\}$ represent a collection of posts, also referred to as a set of textual units. Our summarization algorithm takes $\mathcal{T}$ and an integer k as input, where $\mathcal{T}$ denotes the entire set of textual units and k specifies the desired number of units in the summary. The task is to output a summary $\mathcal{S} \subseteq \mathcal{T}$ such that $|\mathcal{S}| = k$. The summary $\mathcal{S}$ would be evaluated based on its alignment with the preferences of gold standard summarizers. If the context window size of an LLM is W, we assume $\mathcal{T}$ is too large to fit in a single context window.

3.2 Multi-Level Summarization

LLMs have a limited context window, making it impossible to input large text collections all at once within a single window. Consequently, the input must be divided into smaller chunks to perform the desired task (Chang et al. 2024a).

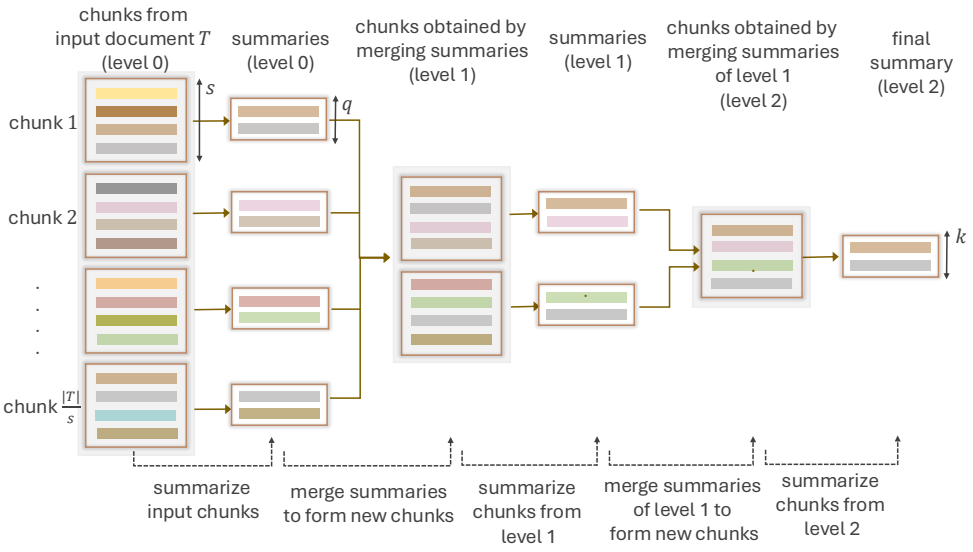

Figure 2: LaMSUM: Multi-level framework for extractive summarization of large user-generated text. Input set $\mathcal{T}$ (level 0) is divided into $\lceil \frac{|\mathcal{T}|}{s} \rceil$ chunks each of size s. From each chunk a summary is produced of size q (refer Figure 3), q length summaries from $\lceil \frac{|\mathcal{T}|}{s} \rceil$ chunks are merged to form the input for the next level i.e., level 1. Iteratively the same procedure is repeated till we obtain a summary of size k. We set $q = k$ to ensure our algorithm can effectively handle the worst-case scenario where all the textual units in the final summary may come from the same input chunk.

Thus, LaMSUM employs a multi-level framework for extractive summarization, enabling it to consider input data of any size (detailed in Figure 2).

The set $\mathcal{T}$, which contains the original textual units, is provided as input at level 0 and is divided into $\lceil \frac{|\mathcal{T}|}{s} \rceil$ number of chunks of size s. From each chunk of size s, we generate a summary of size q (where $q < s$), and repeat this process for all $\lceil \frac{|\mathcal{T}|}{s} \rceil$ chunks.[1] We then merge all these q length summaries obtained from level 0 to form an input for the next level i.e., level 1. We repeatedly perform this process until we obtain the final summary of length k. Note that the last chunk may be less than q in size, in such a case we move all the textual units of the respective chunk to the next level (refer Algorithm 1). Following a long line of research on extractive summarization (Nallapati, Zhai, and Zhou 2017; Liu and Lapata 2019), we have used the summary length as the stopping criterion for our algorithm. Moreover, while evaluating the performance, it is essential for the LLM-generated summary to match the length of the human-written reference summary; otherwise, the ROUGE score would not offer a fair comparison.

An alternative strategy would be to divide the input $\mathcal{T}$ into $\frac{|\mathcal{T}|}{s}$ chunks each of size s and from each chunk select $\frac{k \cdot s}{|\mathcal{T}|}$ sentences to be included in the summary. However, this approach assumes a uniform distribution of potential candidates across chunks that can be included in the final summary. In LaMSUM, we keep $q = k$ i.e., we extract k textual units from each chunk, eliminating the chance of missing any potential candidate. In the worst-case scenario, all k units in the final summary can come from a single chunk, and our algorithm can handle such cases effectively, as we keep $q = k$. Ablation study for different values of q is discussed in Section 5.1.

It is important to note that we are dealing with user-generated posts, which lack contextual connections. Unlike book summarization, where chapters are interconnected and the context of previous chapters is crucial for summarizing the current one, posts are generally standalone and contextually independent. Thus, our approach of independently deriving summaries from each chunk works well in our setup, as each textual unit operates independently of the others and there are no long-range dependencies. However, social media posts can also be contextually connected, including the original post, comments from other users and reposts of others' content. In such cases, our existing framework can be adapted by incorporating clustering at different levels – such as the original post, comments, and comment threads. The LaMSUM framework is extendable enough to be further applied at each hierarchical level.

3.3 Summarizing a Chunk

Next, we discuss how LaMSUM summarizes a chunk (Algorithm 2) by tackling the positional bias in LLMs and leveraging voting algorithms drawn from Social Choice Theory (Brandt et al. 2016).

Tackling Positional Bias: Prior research (Brown and Shokri 2023; Zhang, Liu, and Zhang 2023b; Jung et al. 2019; Wu et al. 2024) has highlighted that summarization using

[1]Note that a chunk of size s refers to a chunk containing s textual units. Likewise, a summary of size q indicates a summary of q textual units. $|\mathcal{T}|$ denotes the number of textual units present in $\mathcal{T}$.

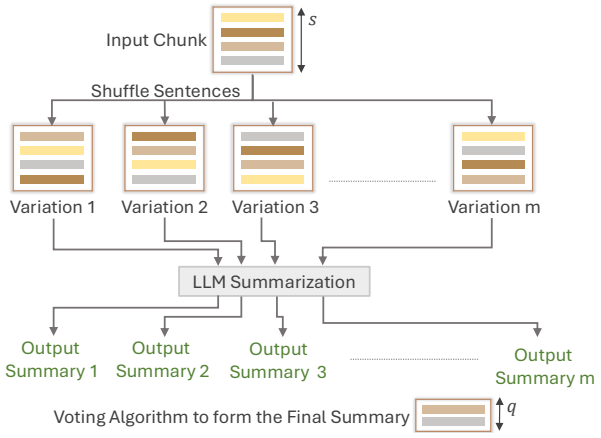

Figure 3: Textual units (e.g., posts) in the input chunk are shuffled to account for the positional bias. m different chunk variations are obtained through shuffling, which are subsequently summarized using LLMs. m summaries are then aggregated by voting algorithms to get the final summary.

LLM is prone to positional bias, i.e., the sentences located in certain positions, such as the beginning of articles, are more likely to be considered in the summary. To address this issue and generate a robust summary, we create m different variations by shuffling the textual units within the input chunk. This ensures that each unit has the opportunity to appear in different positions within the input text (refer Figure 3).

Zero-Shot Prompting: For each input chunk, we obtain m different summaries (one for each variation) by prompting the LLM. We employ the following two prompt strategies to obtain the summaries (detailed in Appendix Table 8):

(1) Select most suitable units that summarize the input text.

(2) Generate a ranked list in descending order of preference.

Output Calibration: LLMs may alter certain words from the input text while generating extractive summaries, as shown in Table 2. Thus, we perform additional checks to ensure that the textual units selected in the summary are indeed a subset of $\mathcal{T}$. If the post selected by the LLM (say x) is not present in the original text $\mathcal{T}$, we identify the post with the closest resemblance to x by computing the edit distance (Ristad and Yianilos 1998). LLMs may also hallucinate, generating new sentences rather than selecting units from the input. In such cases, the edit distance between the generated unit x and the original textual units tends to be high. To address this, we adopt an alternative approach – we extract key elements such as nouns, verbs, and adjectives from the newly generated sentence x. Then we search for the exact same set of keywords with the textual units in $\mathcal{T}$ and identify the post with the highest number of matching keywords as the closest match to x (refer Algorithm 4 in Appendix).

3.4 Reimagining Summarization as an Election

As mentioned earlier, for a given chunk, we obtain m summaries – one for each variation. *We imagine the process of creating the final summary from these m summaries to be a multi-winner election*, where the textual units in m summaries correspond to ballots (candidates) and the role of the voting algorithm is to pick q winners. We employ three different voting methods, namely **Plurality Voting** (Mudambi, Navarra, and Nicosia 1996), **Proportional Approval Voting (PAV)** (Lackner, Regner, and Krenn 2023) and **Ranked Choice Voting** (Emerson 2013) to determine the final summary. Due to the varying input requirements of different voting methods, changing both the prompting approach and the output generated by the LLM becomes imperative.

Plurality voting and proportional voting are approval-based voting methods where voters can select multiple candidates they approve of without indicating a specific preference order. In multi-winner plurality voting (also known as **block voting**), each voter casts multiple votes and the candidates are selected based on the number of votes polled. *In the context of summarization, a textual unit is treated as a candidate, and the LLM acts as the voter*. We select the textual units in the decreasing order of the votes polled, till we obtain a summary of size q. PAV evaluates the *satisfaction* of

Algorithm 1 Algorithm for multi-level summarization

Input: $\mathcal{T}, k, s, q, m$
$\mathcal{S} = \{\}$ ▷ $\mathcal{S}$ stores the final summary
while $|\mathcal{S}| < k$ **do** ▷ until k length summary is obtained
  $n_{chunks} = \lceil \frac{|\mathcal{T}|}{s} \rceil$ ▷ number of chunks in set $\mathcal{T}$
  $L = \{\}$ ▷ L stores the results of a given level
  for $i \leftarrow 1$ to n_{chunks} **do**
    $si = (i-1) * s$ ▷ starting index of chunk
    $ei = i * s$ ▷ ending index of the chunk
    if $i = n_{chunks}$ **then** ▷ if last chunk
      $ei = |\mathcal{T}|$ ▷ ei is equal to length of $\mathcal{T}$
    end if
    $width = ei - si$ ▷ no. of textual units in a chunk
    if $width <= q$ **then** ▷ if last chunk
      $L = L \cup t_{si} \cup t_{si+1} \cup ... \cup t_{ei-1}$
      ▷ add all textual units to the result
    else
      $L = L \cup \text{CHUNKRESULT}(\mathcal{T}, si, ei, q, m)$
      ▷ add summary of each chunk to result L
    end if
  end for
  $\mathcal{T} = L$ ▷ update the input $\mathcal{T}$ for the next level
  $\mathcal{S} = \mathcal{S} \cup L$
end while
Output: $\mathcal{S}$

Algorithm 2 Algorithm for summarization of a chunk

function CHUNKRESULT($\mathcal{T}, si, ei, q, m$)
  $X = \{\}$
  for $i \leftarrow 1$ to m **do** ▷ for each variation of a chunk
    $V = \text{SHUFFLE}(\mathcal{T}, si, ei, i)$ ▷ shuffle with state i
    $R = \text{LLM}(V, q)$ ▷ obtain LLM summary
    $C = \text{CHECK}(R, \mathcal{T}, si, ei)$ ▷ output calibration
    $X.add(C)$
  end for
  return VOTING(X, q) ▷ voting for final summary
end function

Original Post	LLM Modified Output
In a train some people were staring me continuously. It was very uncomfortable.	Some people were staring me continuously in a train.
We were going to metro station, a biker started following us. When we shouted, he rode away.	A biker followed us and rode away when we shouted.
Some boy do dirty comments on me and my religion.	Boys made dirty comments about my religion.

Table 2: Examples illustrating that LLMs when selecting textual units for summarization, often demonstrate a propensity to alter certain words or introduce new ones.

each voter in the election outcome. A voter's satisfaction is measured based on – amongst the number of candidates they voted for, how many are selected in the election. In the realm of summarization, PAV selects the textual units based on the amount of support each unit receives in m summaries. Since both plurality and proportional are approval-based voting algorithms, the units are either approved or disapproved by the underlying LLM, with no explicit ranking or preference order. In this case, we prompt the LLM to *select the best* $< q >$ *sentences that summarize the input text* as shown in (1).

On the other hand, ranked choice voting entails assigning a score to each textual unit and subsequently selecting the highest-scoring units for inclusion in the summary. For ranked voting, we use the Borda Count, a positional voting algorithm (Emerson 2013). In the Borda method, each candidate is assigned points corresponding to the number of candidates ranked below them: the lowest-ranked candidate receives 0 points, the next lowest gets 1 point, and so forth. The candidates with the highest aggregate points are declared as the winners. In ranked voting, we prompt the LLM to *output sentences in descending order of their suitability for the summary* as discussed in (2).

It is important to note that the prompting technique and the output generated by LLM vary for different voting methods. In approval voting, the output from LLM is a list of q textual units that LLM finds best suited to be included in the summary. Whereas in ranked choice voting, the output from LLM is a list of the same length as input i.e. s with all the units sorted in decreasing order of their preference towards the summary, and Borda Count (Emerson 2013) is used to identify the top q textual units. In the next section, we highlight how the voting-based summarization schemes outperform the *Vanilla* setup, which does not use voting.

4 Experimental Setup

This section outlines the experimental setup, detailing the dataset utilized, the LLMs employed for summary generation, and the evaluation metrics used for assessment.

4.1 Dataset

We gather the data from one of the major incident reporting platform in Asia. The dataset comprises incident posts from *five* different cities, denoted as `City A`, `B`, `C`, `D` and `E`. For `City C` and `E`, we obtain the posts for 3 years, i.e., from Dec 2021 to Nov 2024. For `City A`, `B` and `D`, we consider the posts for 5 years, from Dec 2019 to Nov 2024. The rationale behind varying the duration of post selection is to keep the total number of posts below 1000, facilitating more efficient and accurate annotation by the human summarizers. Our dataset of cities `A`, `B`, `C`, `D` and `E` consists of 625, 867, 866, 545 and 728 posts respectively. Sexual harassment can have various categories, such as physical assault, touching, stalking etc. Each post in the dataset is tagged with one or more categories by the author of the post. The distribution of posts across these categories is shown in Table 3. While the platform does not collect personal information such as names or identities, it does gather age, gender, and details of the incidents. Table 9 (in the Appendix) provides an overview of the attributes associated with each post. For our task, we focus solely on the main description provided in the posts.

No.	Post Category	A	B	C	D	E
PC1	Rape/Sexual Assault	33	27	17	23	18
PC2	Chain Snatching/Robbery	49	16	103	32	22
PC3	Domestic Violence	27	87	30	10	34
PC4	Physical Assault	60	33	57	41	33
PC5	Stalking	146	166	165	100	153
PC6	Ogling/Staring	147	100	202	133	209
PC7	Taking Photos	73	70	43	37	55
PC8	Mas*****tion in public	45	55	27	30	42
PC9	Touching/Groping	152	209	206	159	230
PC10	Showing Po**ography	23	10	15	19	15
PC11	Commenting/Sexual Invites	133	192	273	133	131
PC12	Online Harassment	66	43	25	61	33
PC13	Human Trafficking	3	4	2	2	1
PC14	Others	50	23	36	41	16

Table 3: The distribution of posts across each category for five datasets – `City A`, `B`, `C`, `D`, and `E`. Posts tagged with various categories by their authors, with each category representing a different form of sexual harassment.

For each of the five city-specific datasets, we generate gold-standard (reference) summaries created by three domain experts. These experts carefully selected textual units from the posts that are strong candidates for inclusion in the summary. As a result, each dataset has three expert-generated gold-standard summaries, each comprising 50 textual units (posts). The expert annotators were provided with following guidelines for selecting the posts for the reference summary:

1. Diversity: Prioritize diverse posts that represent various forms of assault.
2. Descriptive: Give preference to posts with detailed descriptions over those containing only 2-3 words.
3. Severity: Include posts that depict more serious cases or

require urgent attention.

4. Redundancy: Exclude posts that are repetitive or redundant.

We utilize Fleiss Kappa to measure the inter-annotator agreement (Fleiss 1971). The Fleiss Kappa scores for five datasets – City A, B, C, D and E are 0.550, 0.376, 0.521, 0.543 and 0.446 respectively; showcasing moderate agreement between the annotators. Note that there may be multiple posts describing similar incidents, but different annotators might include different subsets of these posts in their summary. Standard summary evaluation metrics like ROUGE (described in Section 4.3) consider these differences between individual reference summaries by averaging across multiple annotators.

4.2 Large Language Models (LLMs)

LLMs are characterized by their extensive parameter sizes and remarkable learning abilities (Zhao et al. 2026; Chang et al. 2024b). In our work, we utilize two open LLMs: `llama-3.1-8B-instruct` from Meta (Grattafiori et al. 2024), `open-mistral-nemo-2407` from Mistral AI (Mistral AI 2024), and two proprietary LLMs: `claude-3-haiku-20240307` from Claude (Team 2024) and `gpt-4o-mini-2024-07-18` from OpenAI (OpenAI 2024), to conduct experiments.

We focused on using models which are around 8B in parameter size. This choice was guided by practical considerations on the cost of computing infrastructure or API calls: open models of 8B size can conveniently run on GPUs with 40GB VRAM, unlike 70B models which typically require about 140GB VRAM. Similarly, API calls for smaller proprietary LLMs cost roughly one-tenth of their larger versions. For a non-profit platform, which may need to run these algorithms frequently, using larger and more expensive models would likely be impractical.

Across all experiments, we keep temperature, top probability and output tokens as 0, 1.0 and 8192 respectively. The current LLMs offer long context windows but we set the context window length to 8192 tokens to prevent hallucinations and ensure the LLMs adhere to the given instructions. Prior studies indicated that excessively long inputs can increase the likelihood of hallucination in LLM output (Zhang, Liu, and Zhang 2023b). Empirically, we observed that providing long text often led to instruction neglect – such as incomplete responses (e.g., only the first few sentences) or generic placeholders like 'similarly we select other sentences'. In our experiments, limiting the context window to 8192 tokens resulted in minimal hallucination and more consistent compliance with the given instructions.

4.3 Evaluation Metric

To evaluate the quality of summaries generated by LaMSUM, we employ evaluation metrics, namely ROUGE-1, ROUGE-2, and ROUGE-Lsum (Lin 2004). These metrics quantify the degree of overlap between the LaMSUM generated summary and the reference summary, thereby providing a quantitative measure of content preservation and relevance.

Parameters	A	B	C	D	E
#textual units	625	867	866	545	728
#words	20544	12665	23501	22471	20807
#tokens	30816	18997	35251	33706	31210

Table 4: Number of textual units, words and tokens in five datasets.

ROUGE-1 and ROUGE-2 are based on n-gram overlap, where ROUGE-1 measures the overlap of unigrams (individual words), and ROUGE-2 evaluates the overlap of bigrams (consecutive word pairs) between the generated and the reference summaries. These metrics capture the extent to which important words and short phrases from the reference summary are retained in the generated output, thereby reflecting content adequacy at a lexical level.

In contrast, ROUGE-L is based on the Longest Common Subsequence (LCS), which identifies the longest sequence of words that appear in both the candidate and reference summaries in the same order, though not necessarily contiguously. This property allows ROUGE-L to account for sentence-level structure and fluency. Building on this, ROUGE-Lsum extends ROUGE-L by applying the LCS based matching at the sentence level rather than treating the summary as a single sequence. This formulation makes ROUGE-Lsum suitable for extractive summarization tasks, where the generated summary often consists of selected sentences from the source text.

5 Experimental Evaluation

In this section, we present the ablation studies, empirical comparison of LaMSUM with competent baseline models and voting algorithms across datasets. The total number of textual units ($|\mathcal{T}|$), number of words, and number of tokens in each dataset are listed in Table 4. Value of k (length of final summary) is set to 50 for all the experiments. To determine the optimal values of other hyperparameters such as m, s and q, we conduct ablation studies, as discussed next.

5.1 Ablation Study

No. of Shuffles (m) and Chunk Size (s): To identify the optimal number of shuffles m, we performed experiments using three values: 3, 5, and 7. Additionally, we tested two different chunk sizes s, set at 100 and 120. When the chunk size s is 100, out of 60 cases, $m = 3$ gave superior results in 35 cases, $m = 5$ showed best performance in 12 cases, and $m = 7$ produced good results in 13 cases (refer Figure 4). For chunk size $s = 120$, out of 60 cases, $m = 3$, $m = 5$ and $m = 7$ showed good results in 24, 19 and 17 cases respectively (refer Figure 7 in Appendix). As the chunk size grows, more shuffles are needed as fewer shuffles may not sufficiently vary each post's position within the input text. In our later experiments, we chose $m = 3$ and $s = 100$ since this combination consistently delivered the most reliable results.

Which LLM Takes the Lead? We conducted experiments using four LLMs. As illustrated in Figure 5, `claude-3-haiku` outperformed the other three LLMs.

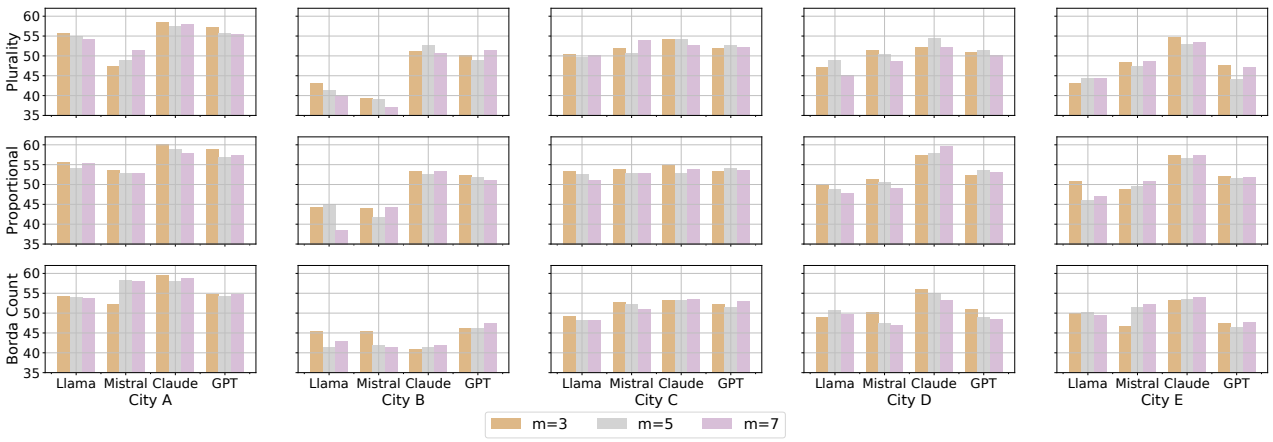

Figure 4: Results for different values of m for chunk size s = 100. It can be observed that in most cases the results of m=3 are either better or comparable to the results of m=5 and m=7. Using m=3 proves to be more efficient for our experimental setup, offering advantages in both time and computational cost.

q	#Levels	ROUGE-1	ROUGE-2	ROUGE-LSum
50	4	60.364	31.988	57.875
60	7	59.940	31.130	57.560
70	9	60.871	30.122	58.276
80	12	60.444	29.676	57.895
90	24	61.132	28.946	58.304

Table 5: Ablation Study for different values of q for `City A` using `claude-3-haiku` model and proportional approval voting for m=3 and s=100.

Summary Size from Chunk (q): When $q \in [k, s)$, our proposed method is capable of handling the worst-case scenario where all textual units in the final summary may originate from a single level-0 chunk. As q approaches s, a greater number of levels are necessary to reach the final summary. We hypothesize that the optimal value of q, which balances handling the worst-case scenario while minimizing the number of levels in multi-level summarization, is k. To validate this hypothesis, we experimented with five different values of q: 50, 60, 70, 80 and 90 on `City A` dataset with Claude model. The results presented in Table 5 indicate that as q increases, the number of levels needed also increases, while the ROUGE scores remain largely unaffected. Paired t-tests revealed that the differences were statistically insignificant, with p-values of 0.085, 0.72, 0.448, and 0.664 when comparing q = 50 against $q = \{60, 70, 80, 90\}$, respectively. Based on these findings, we selected q = 50 as the optimal value, balancing performance and computational efficiency without compromising the results.

5.2 Baseline Comparison

We compare `LaMSUM` with the pre-neural models such as LexRank (Erkan and Radev 2004), SummBasic (Nenkova and Vanderwende 2005) and LSA (Gong and Liu 2001); transformer based models such as BERT (Miller 2019) and XLNET (Yang et al. 2019); state-of-the-art fine-tuned model BERTSUM (Liu and Lapata 2019). Our proposed method, `LaMSUM`, achieves optimal performance when `claude-3-haiku` is used with proportional approval voting. Therefore, we present the results of this combination as the outcome of `LaMSUM`. As shown in Table 6, it is observed that `LaMSUM` surpasses state-of-the-art summarization models across all metrics.

5.3 Does `LaMSUM` Perform Better than Vanilla LLM?

Our proposed framework, `LaMSUM`, ensures robust summary generation by shuffling and employing a voting algorithm to select the best textual units for the summary. It is crucial to compare `LaMSUM` with a multi-level LLM that does not use shuffling and voting, which we call *Vanilla LLM*. Algorithm 3 outlines the steps used by *vanilla LLM* to find the chunk summary. Figure 5 demonstrates that the *vanilla* multi-level LLM has lower ROUGE scores for each LLM compared to the proposed framework `LaMSUM`, indicating that shuffling and voting enhance the performance. Earlier work (Zhang, Liu, and Zhang 2023b) reported that the ChatGPT model achieves lower ROUGE scores on CNN/DM and XSum datasets. Our results demonstrate that our proposed framework performs significantly better than other fine-tuned language models such as BERTSUM for large user-generated text.

5.4 Which Voting Algorithm Performs the Best?

We experimented with three voting algorithms, two approval-based (plurality and proportional) and one rank-based (borda-count). Experimental results indicate that LLMs with proportional approval voting perform the best compared to the other voting algorithms (Figure 5). Differences in performance for different voting algorithms were found to be statistically significant ($p < 0.05$ in paired t-test) for all the setups, except Mistral with plurality based voting and GPT with borda count (refer Table 11 in Appendix).

We hypothesized that rank-based voting would yield better results, as it makes more informed decisions about the

Figure 5: Metric scores obtained through four different LLM setups. (i) Vanilla LLM without shuffling and voting method (ii) `LaMSUM` with plurality approval voting (iii) `LaMSUM` with proportional approval voting and (iv) `LaMSUM` with borda count ranked voting. Results demonstrate that `claude-3-haiku` with proportional approval voting performs the best across all the cases. Here, Llama, Mistral, Claude and GPT refers to `llama-3.1-8B`, `open-mistral-nemo`, `claude-3-haiku` and `gpt-4o-mini` respectively. Error bars represent the standard deviation observed across 3 repeated runs of the same setup.

Models	City A			City B			City C			City D			City E		
	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum
LexRank	57.644	28.084	55.904	38.898	16.909	38.239	56.379	24.897	54.057	46.221	21.709	44.612	52.157	23.681	49.812
SummBasic	52.498	21.848	50.700	26.714	9.536	26.283	53.172	21.600	51.136	43.215	19.841	41.392	50.677	21.357	48.482
LSA	55.250	25.156	53.219	52.529	25.947	51.462	45.623	17.378	43.981	42.370	18.262	40.762	47.522	17.521	45.585
BERT	55.645	23.673	54.236	53.799	26.428	52.711	53.610	22.353	51.982	43.986	19.738	42.322	50.444	21.470	49.377
XLNET	50.559	19.373	48.959	51.706	24.259	50.937	52.819	21.137	51.189	43.768	22.133	42.391	51.650	22.151	50.142
BERTSUM	56.570	25.323	54.226	51.137	24.551	49.386	53.927	22.647	50.993	53.767	25.846	49.376	47.735	20.606	45.295
`LaMSUM`	**62.192**	**33.469**	**59.996**	**53.871**	**32.273**	**53.251**	**57.401**	**29.562**	**54.854**	**59.680**	**36.721**	**57.388**	**59.324**	**37.784**	**57.312**

Table 6: Table showing metric scores from different models for various datasets. Here, R1 = ROUGE-1 Score, R2 = ROUGE-2 Score, RLSum = ROUGE-LSum Score. The best result for each dataset is shown in **bold** and clearly `LaMSUM` outperforms all the other methods across all the evaluation measures.

potential sentences to be included in the summary. Contrary to our expectations, rank-based algorithms did not surpass the proportional approval voting. This can be attributed to multiple factors:

1. LLMs may hallucinate and output sentences in the same or in the reverse order as they were in the input.
2. Occasionally, LLMs do not output all the sentences from the input, resulting in the padding of left-out sentences towards the end of the list, which disturbs the ranking and potentially affects the result.

Proportional Approval Voting (PAV) selects posts in proportion to the support posts receive from different shuffles. Often, several posts express similar ideas – such as repeating a fact, argument, or opinion. There is a risk of redundancy if we select the posts with the highest number of approvals – the summary may emphasize the same point multiple times across different shuffles, while overlooking less frequent but important posts. PAV addresses this by applying diminishing returns. A post gains credit not just based on how many shuffles approved it, but also based on how much utility it provides to each shuffle. For each shuffle, the satisfaction decreases with each additional approved post selected – the first approved post contributes more than the second, the second more than the third, and so on. This approach helps avoid over representation of dominant viewpoints and encourages inclusion of varied content. PAV behaves like an editor that takes into account all shuffles, detects overlaps, and constructs a summary that balances popular content with distinct, less common insight – capturing the collective judgment more effectively.

Algorithm 3 Algorithm for summarization of a chunk in Vanilla LLM

function CHUNKRESULT($\mathcal{T}, si, ei, q, m$)
  $R = \text{LLM}(\mathcal{T}, si, ei, q)$ ▷ q textual units from $[si, ei]$
  $C = \text{CHECK}(R, \mathcal{T}, si, ei)$ ▷ output calibration
  return C
end function

5.5 Analyzing the `LaMSUM` Output

We analyse the difference between the posts selected by the `LaMSUM` and those chosen by the other algorithms. `LaMSUM` selects the posts that are more descriptive and rich in detail, in contrast to posts that lack sufficient information. As shown in Figure 6, it is evident that across nearly all datasets, the posts selected by `LaMSUM` exhibit a higher word count. This indicates that the proposed algorithm is capable of capturing more detailed information compared to the other al-

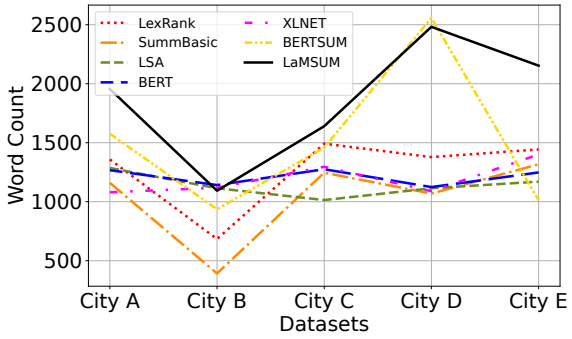

Figure 6: Posts chosen by LaMSUM tend to be detailed and descriptive, offering a deeper level of information. Number of words in LaMSUM selected posts is often highest across various datasets, ensuring extensive and comprehensive summarization.

Models	City A	City B	City C	City D	City E
LexRank	7.486	5.819	7.637	7.548	7.481
SummBasic	8.198	5.734	8.050	8.020	8.196
LSA	8.251	7.061	8.481	8.194	8.387
BERT	8.068	7.762	8.437	8.186	8.191
XLNET	8.144	7.689	8.619	8.205	8.400
BERTSUM	8.331	7.539	8.690	7.960	8.112
LaMSUM	**8.563**	**7.822**	8.622	**8.606**	**8.480**

Table 7: Entropy values representing the diversity in the summaries produced by various algorithms. **Bold** values highlight the best and the underline denotes the second-best performance. LaMSUM achieves the highest diversity score across four datasets.

gorithms (refer Table 13 in Appendix for examples). Furthermore, posts selected by LaMSUM exhibit greater diversity, encompassing a broader range of harassment categories compared to the other baselines. We use entropy as a measure of diversity where a higher value is indicative of more randomness (Jost 2006). Table 7 demonstrates that LaMSUM generates diverse summaries as compared to other baselines.

6 Concluding Discussion

Incident reporting platforms receive numerous posts related to sexual harassment; a summarization algorithm enables end-users to quickly review the most significant ones. This work marks an early attempt to achieve extractive summarization of large user-generated text that exceeds a single context window using zero-shot learning. The proposed multi-level framework LaMSUM leverages approval based and ranked-based voting algorithms to generate robust summaries. Experiments conducted on crowd-sourced dataset demonstrated the efficacy of LaMSUM, as it outperformed the results achieved by state-of-the-art models. While our primary focus was on applying LaMSUM to an incident reporting platform, the proposed framework is generalizable and can be readily used with other social media datasets. We demonstrated this by evaluating LaMSUM on three publicly available datasets – US-Election, Claritin, and Me-Too (Dash et al. 2019). Additional details on the datasets and results can be found in Appendix Section A.5. The overall contribution of this study is a socially grounded framework for extractive summarization of user-reported harassment incidents and the use of LLMs for socially responsible applications.

Note that there can be a concern regarding the potential data leakage, as the experiments involve newer LLMs that may have been exposed to the experimental datasets during their pre-training phase. We demonstrate that the *vanilla LLM*, despite being an LLM-based framework, exhibits inferior performance, whereas our proposed framework generates more robust summaries and delivers improved results. This highlights the efficacy of our model, even when it is exposed to data leakage.

Limitation: Our proposed framework, LaMSUM, very well handles text of any length, conditioned on the fact that the final summary fits within a single context window. Some modifications to LaMSUM may be necessary when the output summary exceeds the size of a single context window.

Ethical Considerations: Our research focuses on using LLMs to produce extractive summaries for incident reporting platforms. LLMs often exhibit bias towards their training data (Chhikara et al. 2024), which can influence their preference for certain textual elements during the summarization process. Their "black box" nature, with an opaque decision-making process, makes it challenging to discern how or why specific textual units are chosen for summarization. The posts selected by LaMSUM may not accurately represent real-life scenarios and cannot serve as a reliable proxy for actual situations. Additionally, LaMSUM may overlook less frequent posts with limited informational content. As a result, exclusive reliance on our framework could lead to the oversight of specific issues by the authorities. While LaMSUM can produce high-quality summaries, its use must be approached with careful consideration of potential ethical implications.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers for their valuable feedback which helped in considerably improving the quality of the paper.

References

Bhattacharya, P.; Poddar, S.; Rudra, K.; Ghosh, K.; and Ghosh, S. 2021. Incorporating domain knowledge for extractive summarization of legal case documents. In *ICAIL*.

Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D. 2016. *Handbook of computational social choice*. Cambridge University Press.

Bražinskas, A.; Lapata, M.; and Titov, I. 2020. Few-Shot Learning for Opinion Summarization. In *EMNLP*.

Brown, H.; and Shokri, R. 2023. How (Un)Fair is Text Summarization?

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell,

A.; et al. 2020. Language models are few-shot learners. *NeurIPS*.

Chang, Y.; Lo, K.; Goyal, T.; and Iyyer, M. 2024a. BooookScore: A systematic exploration of book-length summarization in the era of LLMs. In *ICLR*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024b. A survey on evaluation of large language models. *ACM TIST*.

Chhikara, G.; Sharma, A.; Ghosh, K.; and Chakraborty, A. 2024. Few-Shot Fairness: Unveiling LLM's Potential for Fairness-Aware Classification. arXiv:2402.18502.

Dash, A.; Shandilya, A.; Biswas, A.; Ghosh, K.; Ghosh, S.; and Chakraborty, A. 2019. Summarizing user-generated textual content: Motivation and methods for fairness in algorithmic summaries. *ACM CSCW*.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *ICWSM*.

ElSherief, M.; Belding, E.; and Nguyen, D. 2017. #NotOkay: Understanding Gender-Based Violence in Social Media. *ICWSM*.

Emerson, P. 2013. The original Borda count and partial voting. *Social Choice and Welfare*.

Erkan, G.; and Radev, D. R. 2004. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *Journal of Artificial Intelligence Research*.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*.

FORCE11. 2020. The FAIR Data principles.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*.

Ghosh Chowdhury, A.; Sawhney, R.; Mathur, P.; Mahata, D.; and Ratn Shah, R. 2019. Speak up, Fight Back! Detection of Social Media Disclosures of Sexual Harassment. In *NAACL*.

Gong, Y.; and Liu, X. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *ACM SIGIR*.

Goyal, T.; Li, J. J.; and Durrett, G. 2023. News Summarization and Evaluation in the Era of GPT-3. arXiv:2209.12356.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Hassan, N.; Poudel, A.; Hale, J.; Hubacek, C.; Huq, K. T.; Karmaker Santu, S. K.; and Ahmed, S. I. 2020. Towards Automated Sexual Violence Report Tracking. *ICWSM*.

Jia, R.; Cao, Y.; Tang, H.; Fang, F.; Cao, C.; and Wang, S. 2020. Neural Extractive Summarization with Hierarchical Attentive Heterogeneous Graph Network. In *EMNLP*.

Jin, H.; Han, X.; Yang, J.; Jiang, Z.; Liu, Z.; Chang, C.-Y.; Chen, H.; and Hu, X. 2024a. LLM Maybe LongLM: Self-Extend LLM Context Window Without Tuning. In *ICML*.

Jin, H.; Zhang, Y.; Meng, D.; Wang, J.; and Tan, J. 2024b. A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods. arXiv:2403.02901.

Jost, L. 2006. Entropy and diversity. *Oikos*.

Jung, T.; Kang, D.; Mentch, L.; and Hovy, E. 2019. Earlier Isn't Always Better: Sub-aspect Analysis on Corpus and System Biases in Summarization. In *EMNLP*.

Kanwal, N.; and Rizzo, G. 2022. Attention-based clinical note summarization. In *SAC*.

Kim, S.; Razi, A.; Alsoubai, A.; Wisniewski, P. J.; and De Choudhury, M. 2024. Assessing the Impact of Online Harassment on Youth Mental Health in Private Networked Spaces. *ICWSM*.

Kopackova, H.; and Libalova, P. 2019. Citizen reporting as the form of e-participation in smart cities. In *CISTI*.

Laban, P.; Kryscinski, W.; Agarwal, D.; Fabbri, A.; Xiong, C.; Joty, S.; and Wu, C.-S. 2023. SummEdits: Measuring LLM Ability at Factual Reasoning Through The Lens of Summarization. In *EMNLP*.

Lackner, M.; Regner, P.; and Krenn, B. 2023. abcvoting: A Python package for approval-based multi-winner voting rules. *Journal of Open Source Software*.

Laskar, M. T. R.; Bari, M. S.; Rahman, M.; Bhuiyan, M. A. H.; Joty, S.; and Huang, J. 2023. A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. In *ACL Findings*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

Liu, Y.; and Lapata, M. 2019. Text Summarization with Pretrained Encoders. In *EMNLP*.

Liu, Y.; Shi, K.; He, K.; Ye, L.; Fabbri, A.; Liu, P.; Radev, D.; and Cohan, A. 2024. On Learning to Summarize with Large Language Models as References. In *NAACL*.

Luo, Z.; Xie, Q.; and Ananiadou, S. 2023. ChatGPT as a Factual Inconsistency Evaluator for Text Summarization. arXiv:2303.15621.

Mathew, B.; Saha, P.; Tharad, H.; Rajgaria, S.; Singhania, P.; Maity, S. K.; Goyal, P.; and Mukherjee, A. 2019. Thou shalt not hate: Countering online hate speech. In *ICWSM*.

Miller, D. 2019. Leveraging BERT for Extractive Text Summarization on Lectures. arXiv:1906.04165.

Mistral AI. 2024. Mistral Nemo: Collaborative Innovation with NVIDIA. https://mistral.ai/news/mistral-nemo/. Accessed: 2024-06-22.

Mudambi, R.; Navarra, P.; and Nicosia, C. 1996. Plurality versus Proportional Representation: An Analysis of Sicilian Elections. *Public Choice*.

Mukherjee, R.; Peruri, H. C.; Vishnu, U.; Goyal, P.; Bhattacharya, S.; and Ganguly, N. 2020. Read what you need: Controllable Aspect-based Opinion Summarization of Tourist Reviews. In *SIGIR*.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI*.

Nenkova, A.; and Vanderwende, L. 2005. The impact of frequency on summarization. Technical report, Microsoft Research.

Olteanu, A.; Castillo, C.; Boy, J.; and Varshney, K. 2018. The Effect of Extremist Violence on Hateful Speech Online. *ICWSM*.

OpenAI. 2024. GPT-4o mini: advancing cost-efficient intelligence.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. *NeurIPS*.

Park, H.; and Lee, J. 2021. Designing a Conversational Agent for Sexual Assault Survivors: Defining Burden of Self-Disclosure and Envisioning Survivor-Centered Solutions. In *CHI*.

Pu, X.; Gao, M.; and Wan, X. 2023. Summarization is (Almost) Dead. arXiv:2309.09558.

Ristad, E.; and Yianilos, P. 1998. Learning string-edit distance. *IEEE Transactions on PAML*.

Sambasivan, N.; Batool, A.; Ahmed, N.; Matthews, T.; Thomas, K.; Gaytán-Lugo, L. S.; Nemer, D.; Bursztein, E.; Churchill, E.; and Consolvo, S. 2019. "They Don't Leave Us Alone Anywhere We Go": Gender and Digital Abuse in South Asia. In *CHI*.

Sawhney, R.; Mathur, P.; Jain, T.; Gautam, A. K.; and Shah, R. R. 2021. Multitask Learning for Emotionally Analyzing Sexual Abuse Disclosures. In *NAACL*.

Shin, B.; Floch, J.; Rask, M.; Bæck, P.; Edgar, C.; Berditchevskaia, A.; Mesure, P.; and Branlat, M. 2024. A systematic analysis of digital tools for citizen participation. *Government Information Quarterly*.

Singh, D. D.; Bhattacharjee, R.; and Chakraborty, A. 2025. Rethinking Hate Speech Detection on Social Media: Can LLMs Replace Traditional Models? arXiv:2506.12744.

Stoop, W.; Kunneman, F.; van den Bosch, A.; and Miller, B. 2019. Detecting harassment in real-time as conversations develop. In *Workshop on Abusive Language Online*.

Sultana, S.; Deb, M.; Bhattacharjee, A.; Hasan, S.; Alam, S.; Chakraborty, T.; Roy, P.; Ahmed, S. F.; Moitra, A.; Amin, M. A.; Islam, A. N.; and Ahmed, S. I. 2021. 'unmochon': A tool to combat online sexual harassment over facebook messenger. In *CHI*.

Tam, D.; Mascarenhas, A.; Zhang, S.; Kwan, S.; Bansal, M.; and Raffel, C. 2023. Evaluating the Factual Consistency of Large Language Models Through News Summarization. In *ACL*.

Tang, L.; Shalyminov, I.; Wong, A.; Burnsky, J.; Vincent, J.; Yang, Y.; Singh, S.; Feng, S.; Song, H.; Su, H.; Sun, L.; Zhang, Y.; Mansour, S.; and McKeown, K. 2024. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In *NAACL*.

Tang, L.; Sun, Z.; Idnay, B.; Nestor, J. G.; Soroush, A.; Elias, P. A.; Xu, Z.; Ding, Y.; Durrett, G.; Rousseau, J.; Weng, C.; and Peng, Y. 2023a. Evaluating large language models on medical evidence summarization. *medRxiv*.

Tang, Y.; Puduppully, R.; Liu, Z.; and Chen, N. 2023b. In-context Learning of Large Language Models for Controlled Dialogue Summarization: A Holistic Benchmark and Empirical Analysis. In *NewSumm Workshop*.

Team, A. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku.

Thomas, K.; Kelley, P. G.; Consolvo, S.; Samermit, P.; and Bursztein, E. 2022. "It's common and a part of being a content creator": Understanding How Creators Experience and Cope with Hate and Harassment Online. In *CHI*.

Upadhayay, B.; Lodhia, Z.; and Behzadan, V. 2021. Combating Human Trafficking via Automatic OSINT Collection, Validation and Fusion. In *ICWSM Workshop*.

Venkatasubramanian, K.; Skorinko, J. L. M.; Kobeissi, M.; Lewis, B.; Jutras, N.; Bosma, P.; Mullaly, J.; Kelly, B.; Lloyd, D.; Freark, M.; and Alterio, N. A. 2021. Exploring a reporting tool to empower individuals with intellectual and developmental disabilities to self-report abuse. In *CHI*.

Worledge, T.; Hashimoto, T.; and Guestrin, C. 2024. The Extractive-Abstractive Spectrum: Uncovering Verifiability Trade-offs in LLM Generations. arXiv:2411.17375.

Wu, Y.; Iso, H.; Pezeshkpour, P.; Bhutani, N.; and Hruschka, E. 2024. Less is More for Long Document Summary Evaluation by LLMs. In *EACL*.

Xu, J.; Gan, Z.; Cheng, Y.; and Liu, J. 2020. Discourse-Aware Neural Extractive Text Summarization. In *ACL*.

Yang, X.; Li, Y.; Zhang, X.; Chen, H.; and Cheng, W. 2023. Exploring the Limits of ChatGPT for Query or Aspect-based Text Summarization. arXiv:2302.08081.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Zhang, H.; Liu, X.; and Zhang, J. 2022. HEGEL: Hypergraph Transformer for Long Document Summarization. In *EMNLP*.

Zhang, H.; Liu, X.; and Zhang, J. 2023a. DiffuSum: Generation Enhanced Extractive Summarization with Diffusion. In *ACL*.

Zhang, H.; Liu, X.; and Zhang, J. 2023b. Extractive Summarization via ChatGPT for Faithful Summary Generation. In *EMNLP*.

Zhang, H.; Liu, X.; and Zhang, J. 2023c. SummIt: Iterative Text Summarization via ChatGPT. In *EMNLP*.

Zhang, T.; Ladhak, F.; Durmus, E.; Liang, P.; McKeown, K.; and Hashimoto, T. B. 2024. Benchmarking Large Language Models for News Summarization. *ACL Transactions*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; et al. 2026. A Survey of Large Language Models. arXiv:2303.18223.

Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X. 2020. Extractive Summarization as Text Matching. In *ACL*.

Ziems, C.; Vigfusson, Y.; and Morstatter, F. 2020. Aggressive, Repetitive, Intentional, Visible, and Imbalanced: Refining Representations for Cyberbullying Classification. *ICWSM*.

Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes
   (e) Did you describe the limitations of your work? Yes
   (f) Did you discuss any potential negative societal impacts of your work? Yes
   (g) Did you discuss any potential misuse of your work? Yes
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? NA
   (b) Have you provided justifications for all theoretical results? NA
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA
   (e) Did you address potential biases or limitations in your theoretical framework? NA
   (f) Have you related your theoretical results to the existing literature in social science? NA
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA
   (b) Did you include complete proofs of all theoretical results? NA
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? NA
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? Yes
   (b) Did you mention the license of the assets? Yes
   (c) Did you include any new assets in the supplemental material or as a URL? Yes
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? Yes
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? NA
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? Yes
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA
   (d) Did you discuss how data is stored, shared, and deidentified? Yes

A Appendix

A.1 Algorithm

Algorithm 4 shows the method for output calibration by utilizing two modules – i). minimum edit distance and ii). maximum count of keywords.

Algorithm 4 Algorithm for output calibration

function CHECK($R, \mathcal{T}, si, ei$)
  $Y = \{\}$ ▷ store the result
  for x in R **do** ▷ for each sentence x in LLM result R
    $min\_dist = \infty$ ▷ keep track of min distance
    $min\_idx = -1$ ▷ post with min edit distance
    $max\_count = 0$ ▷ matching keywords
    $max\_idx = -1$ ▷ post with max keywords
    $K = \text{KEYWORDS}(x)$ ▷ obtain keywords in x
    for $i \leftarrow si$ to ei **do** ▷ for each unit in $\mathcal{T}$
      $d = \text{EDITDIST}(x, \mathcal{T}[i])$ ▷ obtain edit distance
      if $d < min\_dist$ **then** ▷ lesser edit distance
        $min\_dist = d$ ▷ update min_dist
        $min\_idx = i$ ▷ update min_idx
      end if
      $c = \text{COUNT}(K, \mathcal{T}[i])$ ▷ # keywords in $\mathcal{T}[i]$
      if $c > max\_count$ **then** ▷ lesser edit distance
        $max\_count = c$ ▷ update max_count
        $max\_idx = i$ ▷ update max_idx
      end if
    end for
    if $min\_dist < \epsilon$ **then** ▷ edit distance is low
      $Y.add(\mathcal{T}[min\_idx])$ ▷ add to result Y
    else
      $Y.add(\mathcal{T}[max\_idx])$ ▷ add to result Y
    end if
  end for
  return Y
end function

A.2 Zero Shot Prompting

Prompts
(1) Select the most suitable units that summarize the input text.
Prompt: Input consists of <chunk_size> sentences. Each sentence is present in a new line. Each sentence contains a sentence number followed by text. You are an assistant that selects best <summary_length> sentences (subset) which summarizes the input. Think step by step and follow the instructions. <sentences>
(2) Generate a ranked list in descending order of preference.
Prompt: Input consist of <chunk_size> sentences. Each sentence is present in a new line. Each sentence contains a sentence number followed by text. You are an assistant that outputs the sentences in the decreasing order of their relevance to be included in the summary. Remember that output should contain all the sentences in the decreasing order of their relevance. <sentences>

Table 8: Prompts utilised for (1) Approval and (2) Ranked-based voting algorithm.

A.3 What Fails to Deliver Extractive Summary?

To ensure extractive summarization, we tested an additional approach – each sentence is tagged with a sentence number, LLM is prompted to *select the best q sentences and output only the sentence numbers of the best q sentences*. Thereafter, the sentences corresponding to the sentence numbers can be retrieved. For instance, if s is 100 and q is 50, the task is to output the sentence numbers of the best 50 sentences from a pool of 100 sentences. In such cases, LLMs hallucinate and provide an output consisting of either all the odd number sentences or all the even number sentences.
Takeaway: For extractive summarization, relying solely on indexes may result in hallucination, underscoring the importance of emitting the input content and not the numbers.

A.4 Post Features

Feature	Details
id	unique id for each post
lang_id	language id
building	building where incident took place
landmark	landmark near the place of incident
area	area where incident occurred
city	city where incident happened
state	name of the state
country	country name
latitude	coordinates information
longitude	coordinates information
created_on	date when the post is made
description	details about the incident
additional_detail	more information about the incident
age	age of the person
gender_id	gender id
gender	gender of the person
incident_date	date when the incident took place
is_date_estimate	binary value – yes or no
time_from	start time of the incident
time_to	end time of the incident
is_time_estimate	binary value – yes or no
categories	harassment category

Table 9: Features or attributes associated with a post. For our task, we utilise only the description feature.

A.5 Generalizability to Other Datasets

We run our experiments on *three* publicly available datasets (Dash et al. 2019). *Claritin* dataset contains 4,037 tweets about the benefits and the side-effects of the anti-allergic drug Claritin. *US-Election* dataset contains 2,120 tweets from 2016 US Presidential Election where people support and attack different political parties. *Me-Too* dataset includes 488 tweets from the October 2018 MeToo movement, where individuals recount the harassment cases they experienced. Results for these datasets are shown in Table 10. Results demonstrate that `LaMSUM` delivers the best results across all baseline algorithms.

Models	Claritin			US-Election			MeToo		
	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum
LexRank	45.04	19.71	44.74	42.63	10.78	41.64	42.70	11.32	40.91
SummBasic	58.25	19.29	56.76	55.36	12.43	53.94	57.23	18.53	54.07
LSA	61.61	23.58	60.74	55.86	15.07	54.81	40.63	11.24	38.86
BERT	57.30	22.37	56.21	55.89	15.44	55.00	45.72	10.76	43.50
XLNET	55.52	21.37	54.75	56.48	15.72	55.41	36.58	08.50	34.48
BERTSUM	57.87	22.75	55.96	59.00	17.51	57.41	57.11	17.08	54.84
LaMSUM	**64.20**	**26.71**	**62.66**	**60.11**	**18.26**	**58.99**	**58.14**	**21.99**	**55.46**

Table 10: Metric scores from different models for various datasets. Here, R1 = ROUGE-1 Score, R2 = ROUGE-2 Score, RLSum = ROUGE-LSum Score. The best value per evaluation measure is shown in **bold** and we can observe that LaMSUM outperforms the baseline models.

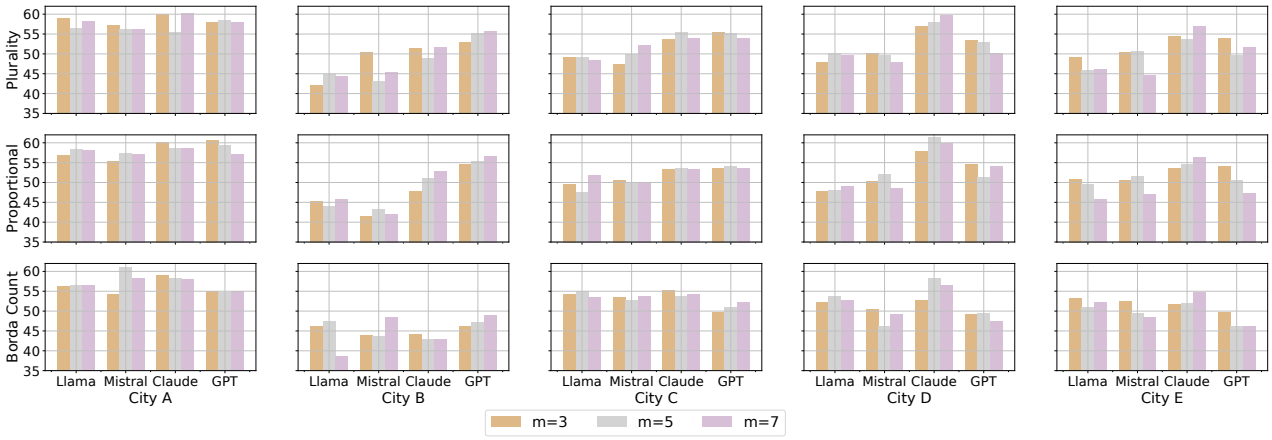

Figure 7: Results for different values of m with a chunk size of $s = 120$ show that out of 60 cases, $m = 3$, $m = 5$, and $m = 7$ achieved good performance in 24, 19, and 17 cases respectively. As the chunk size increases, performing only three shuffles may be insufficient to guarantee that each post occupies varied positions within the input text.

Model	Plurality	Proportional	Borda Count
Llama	0.00061722	0.00000123	0.00003112
Mistral	0.08018935	0.00000118	0.01398864
Claude	0.00011471	0.00000026	0.00855709
GPT	0.00000225	0.00000002	0.07184720

Table 11: The p-values from the paired t-test indicate the statistical significance of the results when comparing the vanilla setup to LaMSUM using various voting algorithms. All comparisons showed statistically significant results with $p < 0.05$, except for for Mistral with plurality voting and for GPT with borda count.

Models	City A			City B			City C			City D			City E		
	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum	R1	R2	RLSum
Vanilla LLM															
Llama	54.860	25.240	52.304	39.391	15.795	38.430	48.935	20.895	46.677	47.528	27.047	45.946	43.416	17.988	41.419
Mistral	52.750	23.234	50.362	40.181	16.885	39.341	52.547	21.972	50.113	51.037	28.340	48.561	42.535	20.544	40.919
Claude	57.913	29.320	55.496	42.209	20.780	41.502	45.103	17.000	42.434	54.525	28.542	51.617	54.413	28.326	51.855
GPT	56.458	25.519	54.214	50.430	26.350	49.008	52.488	22.053	50.336	50.011	24.889	48.246	45.980	20.473	44.439
LaMSUM + Plurality Voting															
Llama	57.876	27.809	55.756	45.459	18.787	43.034	52.376	24.352	50.468	49.407	23.182	47.072	44.879	19.623	43.080
Mistral	50.258	20.426	47.387	40.544	16.201	39.317	54.749	23.093	51.887	53.531	30.556	51.430	51.777	25.503	48.494
Claude	60.896	33.200	58.597	51.807	28.308	51.109	56.950	27.725	54.239	55.130	30.145	52.282	57.361	32.805	54.699
GPT	59.992	26.724	57.271	51.400	26.819	50.183	54.771	22.595	52.002	53.160	26.968	50.944	49.675	23.408	47.548
LaMSUM + Proportional Voting															
Llama	58.136	26.325	55.590	45.844	19.763	44.208	55.884	24.522	53.271	51.655	30.022	49.956	53.669	23.998	50.840
Mistral	56.202	28.285	53.548	45.298	22.138	43.887	56.648	26.570	53.803	55.639	28.618	51.206	51.889	23.516	48.772
Claude	**62.192**	**33.469**	**59.996**	**53.871**	**32.273**	**53.251**	**57.401**	**29.562**	**54.854**	**59.680**	**36.721**	**57.388**	**59.324**	**37.784**	**57.312**
GPT	60.872	31.103	58.844	53.489	29.688	52.356	55.901	26.762	53.179	54.965	29.699	52.366	54.121	27.445	51.981
LaMSUM + Borda Count															
Llama	57.262	25.622	54.349	46.561	21.386	45.518	51.584	23.077	49.278	51.009	28.853	48.961	52.670	24.440	49.949
Mistral	54.928	26.103	52.353	39.762	16.865	38.956	55.995	23.740	52.834	52.379	26.646	50.213	49.389	18.947	46.623
Claude	61.693	30.466	59.470	41.650	16.165	40.822	55.797	25.026	53.268	58.646	35.064	56.132	55.788	28.688	53.387
GPT	57.244	26.107	54.881	47.500	24.153	46.348	55.031	21.72	52.331	52.608	30.263	50.900	49.323	23.184	47.390

Table 12: Table showing metric scores from different LLM models for various datasets. The best value per dataset is shown in **bold** and clearly `claude-3-haiku` with proportional approval voting outperforms all the other methods across all the evaluation measures. In this table, Llama, Mistral, Claude and GPT refers to `llama-3.1-8B`, `open-mistral-nemo`, `claude-3-haiku` and `gpt-4o-mini` respectively. Graphical representation of this table is shown in Figure 5.

Posts
One person whom my family rejected for marriage is posting my nude pictures in social media platforms, hacked my email id and forwarding the nude videos and pictures to all the contacts through different different email ids. Fake Facebook id using my pics and also fake Instagram accounts using my pics, posting bad things about me and my mother. Every day calling and texting me with different numbers. Till date he has taken 10 sim cards. Torturing my family members every day with 40 different numbers. My life has become hell. I have lost my job. Sole bread winner of the family with 2 aged parents 70+ years. Nobody is able to help me.
She was out shopping at a supermarket when she noticed a 35-40 year old man was taking her pictures/videos. Initially, she thought that it might just be a misunderstanding and the man must just be using his phone. But later got too suspicious and scary because he was following her where ever she was going. She even reported the incident to the staff members of that supermarket but before any actions were taken the man had escaped.
I was harassed at my workplace in 2015 at ABC Technology Solutions while working as a Programmer Analyst Trainee by a senior employee in the team who attempted to establish physical contact/advances several times and I was unable to react and I later complained to my reporting manager. Upon complaining to the HR and my reporting managers they claimed that they know rules pertaining to the sexual harassment act and did not take any corrective/legal action and within a few days, I was asked to give forced resignation/termination.
This incident took place around 7 30 in the night. I took a bus home after my college trip and while I was waiting to collect the ticket from the conductor he touched me in my private part in the upper part of my body. I was too confused and scared to speak out something. After a while I got a seat in the bus and was continuously yelled at for sitting crossing my legs and was threatened to be thrown out of the bus. It was late at night and I just wanted to get home safely

Table 13: Posts selected by LaMSUM are more detailed and provide a clearer description of the incident. Detailed posts enable stakeholders to gain a deeper understanding of the incident's context, facilitating more informed and effective decision making.