Comprehensive Abstractive Comment Summarization with Dynamic Clustering and Chain of Thought

Longyin Zhang[1], **Bowei Zou**[1], **Jacintha Wee Yun Yi**[1,2]*, **Ai Ti Aw**[1]
1. Institute for Infocomm Research, A*STAR, Singapore
2. Nanyang Technological University, Singapore
{zhang_longyin,zou_bowei,aaiti}@i2r.a-star.edu.sg
jaci0003@e.ntu.edu.sg

Abstract

Real-world news comments pose a significant challenge due to their noisy and ambiguous nature, which complicates their modeling for clustering and summarization tasks. Most previous research has predominantly focused on extractive summarization methods within specific constraints. This paper concentrates on **C**lustering and **A**bstractive **S**ummarization of online news **C**omments (CASC). First, we introduce an enhanced fast clustering algorithm that maintains a dynamic similarity threshold to ensure the high density of each comment cluster being built. Moreover, we pioneer the exploration of tuning Large Language Models (LLMs) through a chain-of-thought strategy to generate summaries for each comment cluster. On the other hand, a notable challenge in CASC research is the scarcity of evaluation data. To address this problem, we design an annotation scheme and contribute a manual test suite tailored for CASC. Experimental results on the test suite demonstrate the effectiveness of our improvements to the baseline methods. In addition, the quantitative and qualitative analyses illustrate the adaptability of our approach to real-world news comment scenarios.

1 Introduction

With the continued development of handheld devices and the boom of social media, the proliferation of online news is swift and relentless. This has fueled heightened interest among researchers on the topic of public opinion analysis, including tasks like opinion mining (Barker et al., 2016a; Pecar, 2018; Gao et al., 2019; Yang et al., 2019; Huang et al., 2023), sensitive comments detection (Pavlopoulos et al., 2017; Chowdhury et al., 2020; Moldovan et al., 2022; Sousa and Pardo, 2022), and news comments summarization (Khabiri et al., 2011; Dalal and Zaveri, 2013; Aker et al., 2016; Žagar and Robnik-Šikonja, 2021). This study concentrates on the **C**lustering and **A**bstractive **S**ummarization of online news **C**omments (CASC). While the past decade has witnessed substantial progress in text summarization across diverse frameworks (Rush et al., 2015; Liu et al., 2015; Yasunaga et al., 2017; See et al., 2017; Lewis et al., 2020; Wang et al., 2021, 2022; Ouyang et al., 2022; OpenAI, 2023), research specifically targeting CASC remains scarce.

Comment summarization involves extracting a collection of comments either by selecting sentences directly from the comments, referred to as extractive comment summarization, or by compressing the original comments through generative techniques for abstractive summarization. While previous studies have explored a methodology of initially clustering comments into distinct groups followed by summarization (Khabiri et al., 2011; Ma et al., 2012; Llewellyn et al., 2016; Barker et al., 2016b; Pecar, 2018; Žagar and Robnik-Šikonja, 2021), these endeavors have focused on extractive summarization. Our investigation of Reddit comments reveals a spectrum of characteristics influenced by context, platform, and user demographics. In particular, these comments exhibit diversity in style, length, level of detail, and relevance to the contextual discussion, posing significant challenges for summarization using extractive approaches.

In this status quo, this work focuses on abstractive comment summarization with the goal of providing well-organized and extensive coverage of user comments across diverse topics. We confront two main challenges: Firstly, the CASC task, in its nascent phase, suffers from a lack of a formal task definition, an open-source evaluation scheme, and accessible data. This absence largely hinders the expeditious and streamlined evaluation of diverse methods, models, or algorithms. Secondly, although recent Large Language Models (LLMs) have achieved impressive performance in text sum-

*Contribution during the internship at Institute for Infocomm Research.

Findings of the Association for Computational Linguistics ACL 2024, pages 2884–2896

marization (Zhang et al., 2022; Chung et al., 2022; Touvron et al., 2023a; Ouyang et al., 2022; OpenAI, 2023), there are still limitations when applying such LLMs to CASC. On the one hand, online comments on trending topics can be lengthy, surpassing the strict input sequence length constraints of existing foundational models. On the other hand, while existing LLMs excel as text generators, their efficacy in clustering remains underdeveloped.

In this paper, we propose a novel task on abstractive news comment summarization to address the aforementioned challenges. Our contributions encompass three points:

- This paper introduces a straightforward yet effective comment clustering algorithm, coupled with a task-specific ranking mechanism, to dynamically adjust the similarity threshold for comment clustering and rank candidate clusters based on both their density and topical relevance. This method ensures that each comment cluster yields high compactness and discernibleness.

- Based on the clustering algorithm, we propose harnessing moderate-scale LLMs for comment cluster summarization. Particularly, we fine-tune the model through a chain-of-thought manner to first predict aspect terms (ATs) for each comment cluster and then perform AT-based summarization.

- In particular, this work contributes the first annotation scheme and manual test suite tailored for CASC. Notably, different from previous works on topic-based clustering (Khabiri et al., 2011; Ma et al., 2012), we introduce the concept of aspect-based comment clusters to restrict the granularity of each cluster quantitatively.

Quantitative experiments and qualitative analysis of the proposed CASC system demonstrate the effectiveness of our improvements over the baseline systems. Moreover, a real-scenario case study further indicates the practicability of our approach.

2 Related Work

Comment cluster summarization. Previous research on comment cluster summarization can be categorized into three main groups: some solely study comment clustering (Aker et al., 2016; Llewellyn et al., 2016), others focus on summarizing comment clusters in different forms (Dalal and Zaveri, 2013; Barker et al., 2016a; Gao et al., 2019; Huang et al., 2023), while a third group explores both clustering and summarization of news comments (Khabiri et al., 2011; Hsu et al., 2011; Llewellyn et al., 2014; Barker et al., 2016b; Žagar and Robnik-Šikonja, 2021). Our study falls within the latter category that emphasizes both clustering and summarization of comments.

Previous studies of (Khabiri et al., 2011; Ma et al., 2012) are the first to explore extractive summarization of comment clusters, which first group comments into clusters under different topics and then extract comments from each cluster to form the summary. Hsu et al. (2011) alternatively proposed a hierarchical clustering method to summarize YouTube comments based on comment term normalization and key term extraction. Subsequently, Llewellyn et al. (2014) conducted a comprehensive comparison of various clustering methods, including topic-driven clustering, keyword-based clustering, k-means, and cosine distance, demonstrating the superior performance of topic-driven clustering in producing comment clusters suitable for extractive summarization.

However, the above topic-oriented approaches often assume that the resulting summary, derived from topically clustered and extracted comments, aligns with user preferences and presents the crux of the discussed matters. Such assumptions may not always hold. Recognizing this problem, Barker et al. (2016b) conducted a series of data analyses and argued that a good comment summary should not only convey the central issues discussed but also encapsulate the opinions surrounding those issues, suggesting that a good summary should cover finer-grained information beyond just topics. In order to capture more informative comment representations, Žagar and Robnik-Šikonja (2021) proposed to utilize neural sentence embedding methods for extractive summarization of comment clusters. Despite these efforts have been made in comment cluster summarization, all previous works concentrated on extractive summarization, while the abstractive CASC task is largely unexplored. In this paper, we aim to conduct an in-depth study of CASC to fill the gap in this field.

LLM-based summarization. The recent popular LLMs (Zhang et al., 2022; Chung et al., 2022; Touvron et al., 2023a; Ouyang et al., 2022; OpenAI, 2023) have achieved impressive performance in many NLP tasks including text summarization. Notably, the recent paper (Pu et al., 2023) argues

that "summarization is almost dead". In this LLM era, our stance is that summarizing large-scale online user comments remains a challenge even for state-of-the-art (SoTA) foundation models for two reasons: (1) The strict input sequence length limitation of current foundation models makes it infeasible to take all comments as inputs for CASC; (2) Current LLMs are usually pre-trained on text generation tasks, their clustering abilities are far from perfect. Moreover, the quality of text clustering is intricately related to user preference, which is hard to describe using a textual prompt. Our work contributes to building a CASC test suite with user-preferred summaries, which aligns with the new horizons discussed in (Pu et al., 2023).

3 CASC Test Suite

This section introduces the proposed annotation scheme coupled with the manually built test suite for CASC evaluation. Diverging from conventional comment clustering works where the comments are clustered based on the coarse-grained topics, this study focuses more on the finer aspect terms within each news comment. In this paper, we introduce a concept of **C**omment **A**spect **T**erm (*CAT*), which refers to primary objects of the entire discourse, either extracted or abstracted to represent the central focus. The aspect terms serve as the focal points of explicit or implicit opinion expression within the comment. For example, in the following comment, *The current Russian government doesn't represent us, and we definitely don't want a war with our closest people.*, the *CAT*s are determined as *the current Russian government* and *the war*, and the comment is expressing disapproval towards the *CAT*s.

With the above background, we formulate the CASC annotation process in three stages: comment aspect term annotation, comment cluster construction, and comment cluster summarization, as detailed below.

CAT annotation. The process of annotating *CAT* presents two challenges: (1) Since aspect terms can be either extracted or abstracted from the original comment, it is difficult to control the granularity of each annotated *CAT*; (2) User comments usually suffer from unclear reference caused by ellipsis or co-reference, which is hard to avoid when reading each comment independently. The two issues may essentially reduce the annotation consistency.

In this work, we allow the case that each comment has multiple *CAT*s, but we impose a quantitative limit of fewer than 10 words to control granularity. Moreover, to better cover the meaning of each comment, the annotators are instructed to read comments chronologically and refer to the pre-context when labeling aspect terms. The scope of permissible context for reference is limited to a single comment tree (the comment-reply structure). To ensure the quality of annotations and discern optimal candidates, we establish the following rules: (1) Preference is given to *CAT*s directly extracted from the original comment rather than artificially induced ones; (2) When *CAT*s convey the same meaning, we prefer using the same textual expression; (3) Aspect terms are not labeled for comments without opinion expression or those devoid of practical meaning.

As aspect term annotation inherently involves subjective judgment from annotators, we estimate the consistency of *CAT* annotation by comparing the annotations of 200 news comments between two annotators. We approximate consistency by calculating the word-level coverage between each pair of summaries, resulting in a score of 43.67%.[1]

Comment cluster annotation. Manually clustering a large volume of comments from scratch, depending on the limited memory of human annotators, is time-consuming and challenging, fraught with difficulties in ensuring annotation quality. As stated before, we study clustering comments based on aspect terms. Specifically, we consider semantic affinity inferred from the frequency of word co-occurrence between *CAT*s as a reliable indicator of similarity. First, we represent each comment by averaging the GloVe vectors within the manually annotated aspect terms. Then, the comments are clustered based on the cosine distance between aspect term representations through fast clustering with a high similarity threshold of 0.85. Comments lacking manual aspect terms are assigned to a single unique cluster, labeled as `Trivial`. Lastly, annotators manually check the correctness of these clusters according to the following guidelines: (1) Given a cluster $G = \{c_1, \ldots, c_N\}$ with N comments, if the aspect terms and content of c_k significantly differ from others in the cluster, exclude c_k from G and assign it to a suitable comment cluster or the `Trivial` group. (2) If cluster G_i shares a

[1] Given two manually annotated aspect terms after removing the stop words and reducing repetitions, if the word coverage between the two *CAT*s higher than 30% of the averaged word number, we take the two annotated aspect terms as successful coverage.

closely related topic with cluster G_j, combine them into a unified cluster. (3) Check comments annotated with aspect terms in the `Trivial` group to determine if they can be reallocated to an existing non-`Trivial` cluster.

Comment summary annotation. Text summarization annotation has long been contentious, primarily due to the inherent subjectivity of annotators, which can easily influence the standard of summary annotation. Recent studies (Goyal et al., 2022; Zhang et al., 2023; Yang et al., 2023; Pu et al., 2023) have demonstrated that LLM summarizers can perform on par with or better than human summarizers. Among them, the research in (Pu et al., 2023) shows that when quantitatively and qualitatively assessed against human annotators, LLM summaries with increased factuality are notably favored by human evaluators. Inspired by this, we transition from manual summary annotation to LLM-assisted summary annotation. Specifically, based on the manually adjusted comment clusters, we first use GPT3.5 to generate an LLM summary for each comment cluster. Then, we manually post-edit the summaries and refine those suffering from hallucinations to produce the final test suite.

Annotated data overview. Based on comments of news articles sourced from Reddit and New York Times 2017,[2] our annotated dataset contains 11 article clusters (ACs) covering various topics, which involve a total of 20 news articles and 7,958 comments. Guided by our annotation guidelines and with the assistance of four annotators, we constructed 848 comment clusters (CCs) along with their corresponding summaries. Further details are shown in Table 1.

Dataset	Comments (#)	CCs (#)	ACs (#)
DEV	2,532	350	3
TEST	5,426	498	8

Table 1: Statistics for the constructed test suite.

4 Approach

Task definition. Given a group of m news articles under the same topic, $A = \{a_1, \ldots, a_m\}$, the articles contain a group of n comments in total, $C = \{c_1, \ldots, c_n\}$, which comment on the articles from various aspects. Each comment yields a flexible length distribution ranging from a short sentence to a paragraph. The CASC task first groups the n news comments into a set of k comment clusters, $G_1, \ldots, G_k$, and each cluster represents a hot topic discussed by the netizens. Subsequently, a summary is abstracted for each comment cluster represented by $S_1, \ldots, S_k$.

4.1 Comment Clustering

Baseline method. We employ the fast clustering algorithm (Reimers and Gurevych, 2019)[3] as our baseline approach for comment clustering, which offers advantages over classical clustering techniques such as K-Means and Affinity Propagation and provides faster processing times while delivering superior performance. Operating on a cosine-similarity threshold coupled with SBERT sentence representation, it retrieves and ranks local communities with highly similar sentences greedily. Tailored for CASC, this paper seeks improvements to the baseline method from the following two aspects for more robust comment clustering.

Dynamic comment clustering. Our investigation reveals that comment clusters with hot topics tend to attract more user comments, yielding high comment densities, while those tackling less popular topics exhibit relatively sparse clusters. However, the baseline algorithm maintains a static similarity threshold throughout clustering, which struggles to adapt to fluctuating distributions of comment cluster densities. This paper argues that an adaptive dynamic threshold according to the scale of each comment cluster is more practical. Algorithm 1 shows our core clustering algorithm in pseudo-code. As cluster scale increases ($+\Delta$), the similarity threshold increases accordingly, resulting in a higher cluster density. Drawing inspiration from increasing the similarity threshold in the early stage of dynamic clustering to ensure denser comment clusters, this paper proposes a dynamic similarity threshold adjustment process based on a concave function. Formally, the algorithm automatically adjusts the threshold by fitting the following function:

$$\varsigma_t = \sqrt{\mathrm{K}_1 \times (\kappa_t + \mathrm{K}_2)} \quad (1)$$

where κ_t denotes the cluster size, ς_t denotes the similarity threshold, and K_1 and K_2 refer to the hyper-parameters to tune on the DEV set.

Cluster ranking. The vanilla algorithm maintains a static similarity threshold to ensure that the distance between comments within each cluster is

[2] https://www.kaggle.com/datasets/aashita/nyt-comments

[3] https://www.sbert.net/examples/applications/clustering/README.html

below the threshold. However, this algorithm may still lead to redundancy as some comments may be shared across different clusters. Then, the clusters are ranked with the redundant comments filtered out for final clustering. The baseline method ranks candidate clusters solely based on cluster size, which is not comprehensive enough. This limitation is particularly evident in CASC, where user preference is more focused on both topic relevancy and cluster compactness. Different from the baseline method, we argue that cluster density and relevance between the cluster and article background[4] are vital for CASC. With this motivation, we introduce a new ranking mechanism as:

$$R_t = \Sigma_i^n \text{cosine}(c_i, c_t) \times \max(\log(\xi_t + 1), 0.1)$$

where c_t denotes the centroid comment of the t-th cluster, ξ_t denotes the stem word intersection between the centroid comment and the background, and R_t refers to the resulting ranking score. Notably, the first term in the formula can be regarded as the product of the cluster size and the cluster density. Therefore, our ranking scheme considers the cluster size, density, and the correlation between the cluster and background concurrently.

4.2 Summarization of Comment Clusters

With the rapid development of foundation models (Zhang et al., 2022; Chung et al., 2022; Touvron et al., 2023a; Ouyang et al., 2022; OpenAI, 2023), researchers increasingly recognize the effectiveness of LLMs across various NLP tasks. With this inspiration, we study adapting the compact yet effective `Flan_T5_XL` (3B) (abbr. FTX) as a benchmark summarization system to fill the gap of foundation models in CASC.

Instruction-tuning. In line with our human annotation efforts, we aim to enhance the foundation model's capability to summarize each comment cluster by focusing on the aspect terms central to each cluster. With this goal, we design instruction-tuning data in a chain-of-thought style to tune LLMs to extract aspect terms from a comment cluster and then abstract the summary by considering the extracted aspect terms and comment texts. However, manual annotation of such tuning data is labor-intensive and time-consuming. Besides, recent research suggests that current SoTA foundation models perform annotation on par with or better than human (Goyal et al., 2022; Zhang et al., 2023; Yang et al., 2023; Pu et al., 2023). Inspired by this, we automatically build comment clusters from Reddit comments (excluding those in the test suite) and utilize the publicly available `gpt-3.5-turbo` to build the instruction-tuning data. In particular, to strengthen the model's capability in chained aspect term and summary generation, we transform each tuning instance into three forms with hybrid learning goals: (1) Given clustered news comments, tune the LLM model to predict aspect terms for each comment cluster; (2) Given the comment cluster along with the aspect terms that the cluster is centered on, tune the model to generate a summary for the cluster; (3) Given the comment cluster, tune the model to predict aspect terms first and then generate a summary based on the predicted aspect terms and the comments within the comment cluster. In total, we obtain 26,438 tuning instances, consisting of 25,438 instances for tuning and 1,000 for validation.

Hallucination alleviation. Upon manual examination, it was discovered that 28% of the com-

Algorithm 1 Dynamic Fast Clustering

Input: N vectorized news comments C
Output: A list of comment clusters G
Initialization: initial cluster size: κ, initial threshold: ς, ceiling threshold: $thrM = 0.9$
Begin
for t-th comment c_t in C **do**
  scores $\leftarrow$ pairwise_cos_sim(c_t, C)
  scores-k$\leftarrow$ scores.top-k(κ)
  $\varsigma' \leftarrow \varsigma$
  while scores-k[-1] > ς' and $\kappa < N$ **do**
    $\kappa \leftarrow$ Min(N, $\kappa + \Delta$)
    $\varsigma' \leftarrow$ Min($\sqrt{\text{K}_1 \times (\kappa + \text{K}_2)}$, $thrM$)
    scores-k$\leftarrow$ scores.top-k(κ)
  end while
  $G_t \leftarrow$ []
  for s_i in scores-k **do**
    $G_t \leftarrow G_t \cup \{c_i\}$ when $s_i \geq \varsigma'$
  end for
  $G \leftarrow G \cup \{G_t\}$
  Calculate the ranking score R_t
end for
Rank the N clusters in G based on $R_{1...N}$
End

[4]Cluster density refers to the average cosine distance from each data point to the centroid comment of the cluster. In the outer loop of Algorithm 1, c_t is the centroid comment of round t. The background is established by randomly sampling six titles from the article cluster where the comments are located.

ment cluster summaries generated by our fine-tuned LLM model suffer from hallucinations. This phenomenon arises primarily due to the model's inability to grasp the context of the original articles being commented on. To solve this issue, we harness the in-context learning capability of the pre-trained LLM and include background information for comment cluster ranking (see Subsection 4.1) as part of the prompt.

5 Experiments

5.1 Evaluation Metrics

This research evaluates CASC from three aspects: comment clustering, comment cluster summarization, and pipeline evaluation.

Clustering. This work considers both the homogeneity and completeness levels between the gold and predicted clusters by reporting the Normalised Mutual Information (NMI) score $\frac{2\times I(C;L)}{H(C)+H(L)}$ as performance, where C refers to the class labels, L refers to the cluster labels, $H(\cdot)$ is the entropy function, and $I(C;L)$ denotes the mutual information between the two group of labels C and L.

Summarization. For comment cluster summarization evaluation, we take the gold standard comment clusters as inputs to generate summaries. Afterward, we calculate the F1-based Rouge-1, -2, and -L scores (ROUGE-1.5.5) between the generated summaries and the ground truth as performance.

Pipeline evaluation. In the pipeline system, the automatic clustering process could result in mismatches between predicted and standard clusters, making the pipeline evaluation hard to realize. This work introduces a scheme for CASC evaluation. Initially, we define a correct hit[5] as when a predicted summary closely matches the ground truth with a cosine similarity exceeding 0.9 (see examples in Appendix C). On this basis, we report precision (P), recall (R), and F1 scores of the correct hits as the CASC performance.[6]

5.2 Clustering Results

For comment clustering, we compare our approach with the following four baseline methods.

[5]We utilize INSTRUCTOR (Su et al., 2023) for summary representation. It is an embedding model fine-tuned on 330 tasks through instruction-based methods. This model can generate task-aware embeddings by considering both task instructions and input text. To maintain consistency, we follow the original paper to use the instruction "Represent the statement:" to represent each summary for similarity calculation.

[6]For details regarding system settings, please refer to **Appendices** A, B, and C.

Method	NMI
K-Means (MacQueen et al., 1967)	22.61
Affinity Propagation (Frey and Dueck, 2007)	32.04
TC (Llewellyn et al., 2014)	17.92
FC (Reimers and Gurevych, 2019)	32.24
Dynamic FC	32.84
+ `Cluster Density`	33.07
+ `Cluster Density` + `BG`	**33.87**

Table 2: Performance of comment clustering. `BG` denotes the background information. FC and TC are abbreviations for fast clustering and topic-focused clustering.

- K-Means (MacQueen et al., 1967) is a traditional approach that alternatively moves comments to the nearest comment cluster centers and updates these centers accordingly. The value of K, representing the number of clusters, needs to be manually determined.

- Affinity Propagation (Frey and Dueck, 2007) is a clustering algorithm that identifies exemplars or representatives within a set of data points. The algorithm can automatically determine the number of clusters based on the input data.

- Topic-focused clustering (Llewellyn et al., 2014) is one of the most representative works on comment clustering, which comprehensively proves that LDA topic-driven clustering produces better comment clusters for extractive summarization.

- Fast clustering (Reimers and Gurevych, 2019) is the benchmark system that our dynamic comment clustering algorithm sources from. It utilizes a cosine-similarity threshold in conjunction with SBERT sentence representation to avidly retrieve local communities containing sentences with high similarity.

The overall results are presented in Table 2.

The baseline fast clustering method yields better performance than K-Means, Affinity Propagation, and the topic-focused clustering method. Moreover, enhancing the fast clustering method with our proposed dynamic clustering algorithm further improves the performance. On top of the dynamic clustering algorithm, we further evaluate our proposed ranking mechanism in two distinct scenarios: one considering only cluster density and the other incorporating both cluster density and background information. The last two rows show our ranking mechanism improves the proposed dynamic clustering algorithm, with the final system settings achieving the best performance.

Method	R-1	R-2	R-L
FTX (Chung et al., 2022)	19.55	6.40	14.50
FTX + TuneCoT	48.12	20.01	32.73
FTX + TuneCoT + BG	**48.13**	**20.27**	**33.15**

Table 3: Results of comment cluster summarization.

5.3 Summarization Results

Abstractive comment summarization is challenging as it requires more training data to understand and summarize noisy online comments. Due to this limitation, previous works have focused solely on extractive summarization. This work takes the compact yet robust FTX model (Chung et al., 2022) as the baseline abstractive summarizer, which uses only 3B parameters while achieving impressive performance. In particular, we contribute two main enhancements to the baseline system. As shown in Table 3, TuneCoT means employing our designed instruction-tuning instances to fine-tune the baseline model on *CAT* and summary generation in a chain-of-thought style. BG means harnessing the background information extracted from source articles for in-context learning to preserve the contextual integrity of the comment cluster as reflected in the summary. The first two rows show that enhancing the base model with the ability of *CAT* prediction and *CAT*-focused summarization improves performance significantly. Moreover, the last two rows indicate that normalizing the output of the LLM with background further improves performance, a point we will illustrate in Section 6.

5.4 Overall CASC Results

We build the benchmark CASC system by combining the SBERT-based fast clustering (Reimers and Gurevych, 2019) module and the FTX-based summarizer. To conduct a thorough performance evaluation, we additionally experiment with recent popular LLMs, including Llama-7B (Touvron et al., 2023a), Llama2-7B (Touvron et al., 2023b), and ChatGPT (`gpt-3.5-turbo`). The overall results are presented in Table 4.

The first two rows show that our instruction-tuning method significantly improves the overall performance, and Line 3 indicates the proposed dynamic clustering method can further boost the performance. To emphasize the importance of clustering in the CASC task, we conducted an ablation experiment with topic-focused clustering applied. The results in Line 4 demonstrate that such an action leads to a significant drop in performance. Furthermore, the results in the lower part of Table 4 show: (1) Our instruction-tuning data and methods can be well extended to other LLMs to enhance their CASC performance. (2) Our tuned FTX-3B model performs better than the larger Llama models and is comparable to ChatGPT. This finding is meaningful for real-world CASC applications.

Method	P	R	F1
FC + FTX (3B)	26.25	12.00	16.47
FC + FTX (3B)♠	75.41	32.86	45.77
FC† + FTX (3B)♠	**79.58**	54.00	64.34
TC + FTX (3B)♠	12.50	2.14	3.66
FC† + Llama (7B)	46.66	31.86	37.86
FC† + Llama (7B)♠	47.59	32.43	38.57
FC† + Llama2 (7B)	76.32	51.57	61.55
FC† + Llama2 (7B)♠	76.65	53.00	62.67
FC† + ChatGPT (20B+)	79.18	**54.86**	**64.81**

Table 4: Results of the CASC pipeline system. Sign ♠ denotes the model after fine-tuning, while † indicates the use of the dynamic clustering strategy.

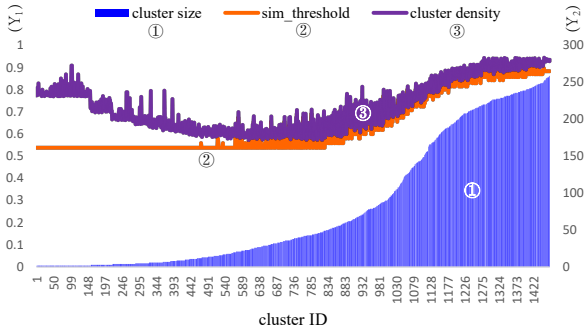

Figure 1: Distribution of similarity threshold and cluster density concerning the comment cluster size. Y_1 and Y_2 refer to the axes with similarity and cluster size scales.

6 Analysis & Discussion

Comment cluster distribution. Figure 1 presents the distribution of 1,467 comment clusters constructed by our CASC system. The column bars labeled with tag ① depict the distribution of comment cluster sizes, the poly-line marked with tag ② denotes the similarity threshold applied to each cluster, and the poly-line labeled with ③ indicates the density value of each cluster. The diagram shows that as the cluster size increases, our dynamic algorithm progressively elevates the similarity threshold, to impose stringent criteria for cluster formation. Since a comment is included in the cluster only when its similarity to the centroid comment surpasses the threshold, the cluster density is distributed above the threshold line, with higher similarity thresholds leading to increased

CMT-1. how do you think you came in contact with flu? **CMT-2**. so, if i caught it, do you think i would know or would i just think it was the regular flu? **CMT-3**. i am currently trying to avoid other humans. although, that sounds like a good revenge for some people. i probably picked up the flu in a public place. **CMT-4**. how would you explain the news describing people with "mild symptoms". do they mean mild compared to regular flu, or mild compared with dying? **CMT-5**. or maybe from a patient who was there for something else and was unknowingly serving as an incubater for the disease. excuse me if you've already answered this, but were you not given every possible vaccination, considering your profession? and even if none are effective, are health-care workers continuing to take every precaution against infection of this flu, or have things been a bit laxer lately? do you mind giving the region of texas in which you work?

Auto aspect terms: flu, contact, symptoms, mild symptoms, vaccination, health-care workers.

Summary w/o BG: The user comments revolve around various aspects related to the **COVID-19 pandemic**. Users express concerns about how they may have contracted the flu and how they would know if they had it if they caught it. They also discuss symptoms and severity levels of the virus as well as health-care worker precautions taken against it. Some users speculate about potential incubators of the disease such as patients who may have served as incubators for others. Overall, there is a focus on understanding how best to prevent the flu.

Summary w/ BG (H1N1 virus, swine flu): The user comments revolve around various aspects related to the **H1N1 virus and swine flu**. Users express concerns about how they may have contracted the flu and how they would know if they had it compared to regular flu symptoms. They also discuss how mild symptoms are interpreted and whether they refer to a comparison with dying or if they indicate an unknowingly serving as an incubater for the disease. Additionally, users question whether health-care workers are taking proper precautions against infection and if things have become laxer.

Figure 2: Example summaries generated by our CASC system for the comment cluster with a gray background.

cluster density. Upon reviewing the real data, it becomes evident that popular topics tend to attract more comments within the cluster, while smaller clusters usually revolve around unpopular topics triggered by the leading comment. In this situation, our method raises the threshold to maintain the integrity of larger clusters and assigns lower threshold values to smaller ones to encourage diversity.

Factual error analysis. LLMs enriched with dense knowledge exhibit heightened creativity but also entail risks of factual inaccuracies. Figure 2 presents an example processed by our CASC system, where all the comments are from Reddit under the topic of "swine flu". We observe that the aspect terms encompass most key points expressed in the cluster and are also embodied in the summaries. The results indicate that by establishing a coherent chain of thought for generating aspect terms and summaries, our method mitigates factual errors to some extent. Nevertheless, the first summary (w/o BG), which revolves around the COVID-19 pandemic, is inconsistent with the topic discussed within the original comments. This is due to an information gap, as the pre-trained LLM model lacks specific knowledge about the themes of the articles that the comments are focusing on. In contrast, after manually incorporating the background knowledge "H1N1 virus, swine flu" into the prompt context, the second summary correctly references the background information, thus avoiding factual errors.

To quantitatively analyze the impact of the background information incorporated, we further conduct CASC on Reddit comments of ten randomly selected article clusters in real scenarios. On this basis, we perform a manual factuality evaluation based on the summaries of the first five longer comment clusters among each article cluster (fifty summaries in total). For the evaluation guideline, we adhere to the rule that a comment cluster summary is deemed factually accurate only when all the entities, events, and opinions it encompasses are entirely in line with the content of the original comments. The evaluation results in Table 5 demonstrate that utilizing background in our final system significantly enhances its factuality.

Method	Factual Error
Final System	18%
w/o BG	28%

Table 5: Manual evaluation of summarization factuality.

7 Conclusion

This paper presents an initial exploration into Clustering and Abstractive Summarization of online news Comments (CASC). We introduce a manual annotation scheme, create a test suite, and propose an evaluation metric tailored for the CASC task. On this basis, we introduce a dynamic clustering algorithm and a novel ranking mechanism for effective comment clustering. In particular, we perform the first practice of tuning foundation models for comment cluster summarization. To foster further studies, the data and codes will be released to the research community upon application.

8 Limitations

As an early exploration in CASC, our study faces two notable limitations: (1) Our annotation efforts have been confined to English news comments due to constraints in human resources. In future research endeavors, we will aim to broaden our resources to construct a multi-lingual CASC test suite. (2) This paper establishes a baseline CASC system using LLM. Although we utilize chain-of-thought fine-tuning along with the in-context learning of background information to alleviate the hallucination issue, the strategy is not thorough enough. We envision undertaking task-specific re-training of LLMs in future work to address this issue further. (3) Currently, the pipeline method outlined in this paper does not fully leverage LLMs. Given the rapid advancement of foundation models, further investigation is needed to explore methods that bridge the gap between clustering and summarization and the capacity to learn clustering and summarization using LLMs simultaneously. (4) When constructing the test suite, we considered the comment tree a context when performing manual annotation, but our system does not consider the comment tree. Inspired by previous work (Tan et al., 2022), exploring discourse theory within comment trees could help with more intelligent news comment analysis in future work.

9 Acknowledgments

We thank Tharini De Silva, Yang Ding, Tarun Kumar Vangani, and other team members for their useful discussions and contributions to the data construction. We express our gratitude to the anonymous reviewers who highly rated this paper and offered insightful comments that significantly contributed to the enhancement of this work.

References

Ahmet Aker, Monica Paramita, Emina Kurtic, Adam Funk, Emma Barker, Mark Hepple, and Rob Gaizauskas. 2016. Automatic label generation for news comment clusters. In *Proceedings of the 9th International Natural Language Generation Conference*, pages 61–69. Association for Computational Linguistics.

Emma Barker, Monica Lestari Paramita, Ahmet Aker, Emina Kurtić, Mark Hepple, and Robert Gaizauskas. 2016a. The sensei annotated corpus: Human summaries of reader comment conversations in on-line news. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 42–52.

Emma Barker, Monica Lestari Paramita, Adam Funk, Emina Kurtić, Ahmet Aker, Jonathan Foster, Mark Hepple, and Robert Gaizauskas. 2016b. What's the issue here?: Task-based evaluation of reader comment summarization systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3094–3101.

Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Abdelali, Soon-gyo Jung, Bernard J Jansen, and Joni Salminen. 2020. A multi-platform arabic news comment dataset for offensive language detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6203–6212.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Mita K Dalal and Mukesh A Zaveri. 2013. Semisupervised learning based opinion summarization and classification for online product reviews. *Applied Computational Intelligence and Soft Computing*, 2013:10–10.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Shen Gao, Xiuying Chen, Piji Li, Zhaochun Ren, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. Abstractive text summarization by incorporating reader comments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6399–6406.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Chiao-Fang Hsu, James Caverlee, and Elham Khabiri. 2011. Hierarchical comments-based clustering. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 1130–1137.

Yuxin Huang, Shukai Hou, Gang Li, and Zhengtao Yu. 2023. Abstractive summary of public opinion news based on element graph attention. *Information*, 14(2):97.

Elham Khabiri, James Caverlee, and Chiao-Fang Hsu. 2011. Summarizing user-contributed comments. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pages 534–537.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training

for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Clare Llewellyn, Claire Grover, and Jon Oberlander. 2014. Summarizing newspaper comments. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, pages 599–602.

Clare Llewellyn, Claire Grover, and Jon Oberlander. 2016. Improving topic model clustering of newspaper comments for summarisation. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 43–50.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 265–274.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Andreea Moldovan, Karla Csürös, Ana-Maria Bucur, and Loredana Bercuci. 2022. Users hate blondes: Detecting sexism in user comments on online romanian news. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 230–230.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*.

Samuel Pecar. 2018. Towards opinion summarization of customer reviews. In *Proceedings of ACL 2018, Student Research Workshop*, pages 1–8.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rogério Sousa and Thiago Pardo. 2022. Evaluating content features and classification methods for helpfulness prediction of online reviews: Establishing a benchmark for portuguese. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 204–213.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Xin Tan, Longyin Zhang, Fang Kong, and Guodong Zhou. 2022. Towards discourse-aware document-level neural machine translation. In *IJCAI*, pages 4383–4389.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022. Clidsum: A benchmark dataset for cross-lingual dialogue summarization. *arXiv preprint arXiv:2202.05599*.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. 2021. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021*

Conference on Empirical Methods in Natural Language Processing, pages 8696–8708, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sen Yang, Leyang Cui, Jun Xie, and Yue Zhang. 2019. Making the best use of review summary for sentiment analysis. *arXiv preprint arXiv:1911.02711*.

Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Aleš Žagar and Marko Robnik-Šikonja. 2021. Unsupervised approach to multilingual user comments summarization. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 89–98.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

A System Settings

Our system is based on the `Flan_T5_XL` (Chung et al., 2022) model with 3B parameters. We tuned all the model parameters for 3 epochs using our built 26,438 instruction-tuning instances. We implemented the CASC system using the PyTorch framework. We tuned our model on A40 GPU cards, and a complete model tuning took around 10 GPU hours. All tests were finished in a single run with the random seed 19. We tuned the hyper-parameters like the number of titles used as background (Table 6), the Δ value (Table 7), the cluster number of the K-Means algorithm (Table 8), the parameters of the curve in Equation 1 (Appendix B), etc., based on the DEV set.

B Tuning Clustering Curve Steepness

Our proposed dynamic clustering algorithm maintains two hyper-parameters in Equation 1 to shape the curve to control the dynamic similarity thresholds. Figure 3 illustrates an example curve. The curve steepness is determined by the two hyper-parameters K_1 and K_2, which are tuned on DEV by sampling data points with the cluster sizes κ_1=50 and κ_1=200 on the X-axis. The two parameters are determined by the empirical investigation that a comment cluster with 50 comments is considered a common size, so we need to establish a general similarity threshold ς_1 to ensure that the comments within the cluster are identifiable; a cluster with 200 comments is deemed long-winded, where we need a higher threshold ς_2 to strictly filter out those unrelated comments to make the cluster more readable. To achieve the above goal, we first assume the curve to be a plain horizontal line to test the effects when ς_1 takes different values on the Y-axis. Table 9 shows that as the value of ς_1 grows, the performance drops sharply. Therefore, we obtain ς_1=0.52 with the highest NMI score on DEV. Then we anchor (κ_1, ς_1) to tune the data point (κ_2, ς_2), where the higher the value of ς_2, the steeper the curve. It shows that it obtains the best performance when the similarity ς_2 is 0.8 for the cluster with 200 comments.

Title Number	R-1	R-2	R-L
0	45.35	16.59	29.13
2	45.22	16.61	29.27
4	45.24	16.49	29.37
6	45.20	16.67	**29.41**
8	45.22	16.80	29.37

Table 6: Title number determined on DEV.

Δ	NMI
5	67.28
10	**67.38**
20	67.35
40	**67.38**
80	67.17

Table 7: Δ value determined on DEV.

C Clustering Speed

The section presents the speed of the fast clustering algorithm, K-Means, and Affinity Propagation for reference, as shown in Table 10. Here, the K-Means and Affinity Propagation algorithms we use are implemented by scikit-learn[7]. Notably, the running speed fluctuates greatly with changes in the total number of comments.

[7] https://scikit-learn.org/stable/modules/clustering.html

K-Means	NMI
2	16.08
4	25.35
8	**38.16**
16	27.97
32	15.52
64	14.43
128	15.74

Table 8: K value determined on DEV.

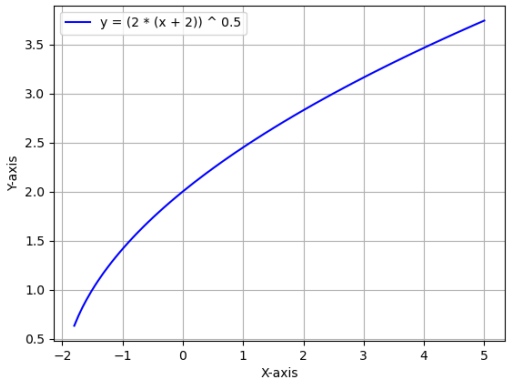

Figure 3: Example curve with K_1=2 and K_2=2.

κ_1	ς_1	NMI	κ_2	ς_2	NMI
50	0.48	64.91	200	0.60	66.65
	0.52	**65.33**		0.65	66.84
	0.56	42.32		0.70	66.94
	0.60	35.40		0.75	67.12
	0.64	13.61		**0.80**	**67.38**
	0.68	11.31		0.85	67.25

Table 9: Curve steepness tuning on the DEV set.

Method	Speed
K-Means	228 ms / 512 comments
Affinity Propagation	265 ms / 512 comments
Fast Clustering	86 ms / 512 comments

Table 10: Speed of different clustering algorithms.

D Successful Hit Summary Pairs in the Pipeline Evaluation

As stated in the main body, this research proposes the pipeline evaluation scheme tailored for the CASC task where we set the cosine similarity threshold to 0.9 to strictly evaluate the quality of the generated summaries. This section presents some summaries that successfully hit the gold standard summaries for reference.

D.1 Example 1

Ground truth. The user comments discuss the impact of high cost of living on the poorest in society and how they cannot switch to cheaper brands or travel less. They also criticize msm for claiming that high cost of living has little effect on the poor. Some users express frustration with an article that talks about the poor only focusing on wealthy individuals. There is a call for increasing GST to help the poor while others suggest taking half of mayors' salaries to subsidize them directly.
Model prediction. The user comments discuss the high cost of living in Singapore and its impact on the poor. Some users criticize the choice of interviewee for the article, questioning whether it accurately represents the struggles faced by those living in poverty. Others point out that this issue is not unique to Singapore and highlights the need to reduce the gap between rich and poor. There is also discussion about supermarket chains like ntuc fairprice not passing on higher prices to consumers despite being strategically placed. Overall, there is a sense that rising prices are causing hardship for those living in poverty.

D.2 Example 2

Ground truth. The user comments revolve around the topic of defining and calculating the poverty line. Some users suggest using minimum income standards as a starting point while others argue that every generation's poverty line is different. There is also discussion about household division and how it can be too low below sustainable levels. The UK's definition of poverty is 60% below median household income after housing costs which some people dislike as it is a relative measure rather than an absolute measure of poverty. Some users suggest using the average wage as a metric but others argue that doing so would include irrelevant people into the calculations, making the metric useless.
Model prediction. The user comments discuss different approaches to defining poverty, including the UK's 60% below median household income after housing costs and the living wage as a personal income measure. They also mention the UK's state pension system and minimum income standards for defining the poverty line. One comment suggests that Singapore should have an official poverty line defined by the bottom 0.1%, which may result in fewer poor people.

D.3 Example 3

Ground truth. The user comments discuss the high number of people being helped by food banks since COVID. One user questions if this is truly

Comment Cluster

1. [Aspect_Terms: poor line, poor people] Since we don't have an official poverty line, probably because that way we have less poor people. If we define the bottom 0.1% to be poor, we will have very few poor people.
2. [Aspect_Terms: poor line] Tbh I've been working on deriving the poverty line. the idea is that every generation poverty line is different.
3. [Aspect_Terms: poor line] The minimum income standards are a good start at trying to define the poverty line.
4. [Aspect_Terms: poor line, household division] The household division makes sense, total income divided by pax but the number here is just way too low below sustainable.
5. [Aspect_Terms: poor line, metric] Yes, the larger the gap is, the less accurate the metric become
6. [Aspect_Terms: poor line, UK] The UK definition of poverty is 60% below median household income after housing costs. So that's a household approach and is used to calculate how many live in poverty. The right don't like it because it is a relative measure rather than a measure of absolute poverty.
7. [Aspect_Terms: poor line, average wage] The average wage should be the poverty line tbh.
8. [Aspect_Terms: poor line, average wage] I don't think the average wage should be poverty line, that would mean like more than 50% of people are poor, inclusive of rich tai tais, stay at home parents etc and all children. It will become a useless metric if too many irrelevant people are included.

Extractive Summary

1. Since we don't have an official poverty line, probably because that way we have less poor people. If we define the bottom 0.1% to be poor, we will have very few poor people.
2. The minimum income standards are a good start at trying to define the poverty line.
3. The household division makes sense, total income divided by pax but the number here is just way too low below sustainable.
4. The average wage should be the poverty line tbh.
5. I don't think the average wage should be poverty line, that would mean like more than 50% of people are poor, inclusive of rich tai tais, stay at home parents etc and all children. It will become a useless metric if too many irrelevant people are included.

Abstractive Summary

The user comments revolve around the topic of defining and calculating the poverty line. Some users suggest using minimum income standards as a starting point while others argue that every generation's poverty line is different. There is also discussion about household division and how it can be too low below sustainable levels. The UK's definition of poverty is 60% below median household income after housing costs which some people dislike as it is a relative measure rather than an absolute measure of poverty. Some users suggest using the average wage as a metric but others argue that doing so would include irrelevant people into the calculations, making the metric useless.

Figure 4: Example comment cluster annotated with extractive and abstractive summaries.

happening while another explains that food banks are used to providing support for those in the poorest 10% of the population but are now supporting more people. It is noted that many families are close to needing their help due to unexpected events such as illness or job loss.

Model prediction. The user comments discuss the claim that Singapore's food banks are helping 1 million people since COVID-19. However, one user questions if this is truly the case and suggests that it may be more accurate to say that the poorest 10% of the population is now needing help. Another user points out that many families are close to needing assistance due to unexpected events like illness or job loss. Despite this, there are still over 30,000 households reliant on foodbanks pre-pandemic and over 100k registered with foodbank.sg.

E Abstractive *vs.* Extractive Comment Summarization

In addition to the final dataset presented in the main body of this paper, our initial exploratory research extends to the extractive summarization of news comment clusters. In extractive summarization annotation, the annotators are guided to manually extract comments from the original comment cluster following two rules: (1) Since each comment is manually annotated with aspect terms, the summary content should cover all the annotated aspect terms; (2) Each aspect term could be shared among multiple comments, the annotators should read and select the most representative comments as the summary to avoid duplication and redundancy. With this guideline, we annotated summaries for 6 article clusters. For reference, we present a comment cluster with extractive and abstraction summaries annotated in Figure 4. When compared, extractive summarization proves superior in accuracy, fluency, and conciseness. This showcases the adaptability of LLM-based summary generation to noisy real-world data.