Controllable Clustering with LLM-driven Embeddings

Kerria Pang-Naylor[1], Shivani Manivasagan[1], Amy Zhong[1], Mehak Garg[1], Nick Mondello[1], Blake Buckner[1], Jonathan P. Chang[1], Khyati Mahajan[2], Masoud Hashemi[2], Fabio Casati[2,3]

[1]Harvey Mudd College, [2]ServiceNow, [3]University of Trento

Correspondence: jpchang@hmc.edu, fabio.casati@servicenow.com

Abstract

Given the inherent subjectivity of similarity in text, fully unsupervised text clustering is unlikely to produce groupings that are relevant across a variety of use cases. Traditional techniques to guide clustering rely on costly, time-consuming human feedback and/or pre-existing labels. Leveraging recent advancements in LLMs and decoder-only embedding models, we present techniques to effectively control text embeddings with minimal human input: instruction prefixing and LLM preprocessing. We evaluate clustering performance for datasets with multiple independent ground-truth labels, or perspectives, and find that these techniques can be used to improve clustering for one perspective or use case, at the cost of a tradeoff in performance for another use case.

1 Introduction

Clustering is a central component of enterprise process analysis. For example, in IT Service Management (ITSM), common asks by Process Owners (POs) include "What are the most common user complaints?", "What are the underlying causes of problems?", and "Why are tickets rerouted to agents?". The answer POs seek involve grouping complex information into buckets *that make sense from one's analysis perspective*. The analysis typically involves large and complex datasets that span thousands of cases. The way POs seek to make sense of this information is through clustering it, and indeed nearly any enterprise software vendor supports clustering.

Clustering aims to group similar data points; however, *similarity* for data points often depends on the analyst's perspective. Consider a dataset of social media posts: a content moderator might want to group posts with similar emotional tones and obscene language indicators (Oskouie et al., 2024; Kumar et al., 2023), whereas a medical researcher might look for groupings based on mentions of drug use or illnesses, irrespective of sentiment (Yang et al., 2019; Chan et al., 2025). Each of these analytical perspectives produces distinct yet equally valid ways of organizing the same data.

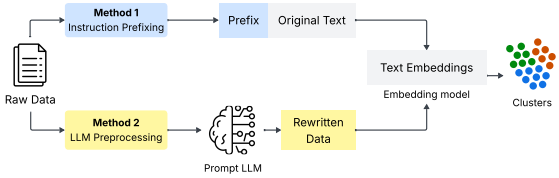

Figure 1: We explore two primary techniques to inject user perspective: *instruction prefixing* (top) and pre-embedding text transformation with *LLM preprocessing* (bottom).

We therefore argue that unsupervised clustering is an **ill-defined problem** (Caruana et al., 2006), and that clustering is inherently subjective. In a space of many possible "correct" groupings, a generic clustering algorithm is unlikely to organize data in a way relevant to a user's specific use case (Caruana, 2013).

Traditional methods that offer more control over clustering include semi-supervised clustering (Butyaev et al., 2022; Qin et al., 2019) and meta clustering (Caruana et al., 2006). However, both techniques have significant limitations. Semi-supervised clustering relies on expensive human feedback or annotation, which limits scalability (Viswanathan et al., 2023). Traditional meta clustering is incompatible with abstract, high-dimensional data with correlated features, such as text representations (Caruana et al., 2006; Dasgupta et al., 2012).

In this paper, we explore an alternative approach for providing control over clusters: we leverage recent advances in instruction-tuned LLMs (Nie et al., 2024; Cao, 2024; Tao et al., 2024) to manipulate the embedding space itself via prompting, while keeping the clustering algorithms un-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 686–702

changed. Specifically, we aim to answer the following research questions:

RQ1: Can embedding modification techniques control clusters to favor a user-specified perspective which may be underrepresented in baseline unsupervised clustering?

RQ2: How much *a priori* information is needed to achieve this control?

Using short, intuitive prompts, we show that perspective-injection techniques can reshape embedding spaces to reflect various viewpoints without relying on labor-intensive supervision methods. We evaluate this approach on a variety of multi-perspective datasets representing real-world use cases, and demonstrate a generalizable ability to reveal patterns that would have been overlooked by traditional unsupervised clustering.

2 Background and Approaches

Most existing work on incorporating LLMs into text clustering has leveraged their powerful text generation capabilities. LLMs can simulate expert feedback to guide clustering (Viswanathan et al., 2023; Zhang et al., 2023; Yang et al., 2024; Trivedi et al.), replacing the costly human input required in traditional semi-supervised clustering. However, scalability becomes an issue as dataset size increases. LLMs can also be applied post-hoc to improve interpretability of text clustering, by generating labels or freeform explanations for clusters (Nie et al., 2024).

Several modern LLM-based text clustering systems such as NV-Embed (Lee et al., 2024) and LLM2Vec (BehnamGhader et al., 2024) include an optional prompting step, and can be used as text embedders. Their key advantage over traditional embedding methods is that instruction-tuned LLMs are responsive to prompting, enabling zero-shot instructional control over the embedding space. We utilize this property for our study.

2.1 Our Contribution: Injecting Perspective

Existing work has largely explored how LLMs can improve clustering performance on a single predefined set of ground-truth clusters per dataset. To our knowledge, we are the first to explore the complementary question of how LLMs can be used to inject *multiple* alternative perspectives per dataset. To this end, we explore both generation-based and prompting-based methods for injecting perspective into the clustering pipeline.

Pre-embedding Text Transformation. One way to inject a desired perspective into the clustering pipeline is to transform the input text prior to embedding to better align with that perspective. Generic summarization has been shown to be ineffective for this purpose (Petukhova et al., 2025). Outside of clustering, however, prompting an LLM to perform a guided transformation (focusing on or excluding specific properties) has improved performance in other downstream tasks (Hua et al., 2024; Chang et al., 2024; Li et al., 2023). We apply this *LLM preprocessing* concept by using an LLM to rewrite text before embedding, investigating both **inclusion** (retain only relevant information) and **exclusion** (remove confounding information) prompts.

Instruction Prefixing. We explore how *prompting* influences the behavior of instruction-tuned LLM embedders. Prompting typically involves prepending an instruction to the input text, a process we refer to as *instruction prefixing*. Traditionally, such prefixing has been used to better align the embeddings with tasks in standard evaluation benchmarks like the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022), thereby improving performance.

Our use of prefixing is different (and, to our knowledge, novel): rather than aiming to improve performance on a given ground-truth label or metric, we seek to reshape the embedding space to discover alternative ways to cluster the same data. We achieve this by systematically changing the prefix (e.g., "Cluster by topic" vs. "Cluster by sentiment").

3 Methods

Our experimental design evaluates how effectively instruction-based techniques can shape embeddings to align with user-defined perspectives during clustering. We design multiple prompt variations for both LLM preprocessing and instruction prefixing, and evaluate them on three datasets representing a diverse range of real-world use cases. To support reproducibility, we use open datasets related to our primary use cases of interest: customer support and developer support in enterprise workflow automation.

3.1 Datasets

A critical requirement for our study was the availability of datasets with multiple, orthogonal

ground-truth labels for the same documents. This allows us to quantitatively measure how well an instruction can steer the clustering outcome toward one perspective (e.g., topic) and away from another (e.g., sentiment). We used modified subsets of the following publicly available datasets:

- **Customer Support** (Bitext, 2024): 1000 customer support interactions, labeled by *inquiry topic* (4 labels, e.g., financial, account) and *sentiment* (positive, rude). We conduct most subsequent analysis on this data due to its applicability in real-world scenarios.
- **StackOverflow** (Shah et al., 2024): 414 programming questions, labeled by *programming language* (4 labels) and *question intent* (debugging, implementation, conceptual).
- **Coarse Discourse Subreddit** (Zhang et al., 2017a): 10,318 posts labeled by *subreddit* (5 labels representing topic) and *comment speech type* (10 labels, e.g., elaboration, appreciation).

All datasets were filtered to include only instances with the specified labels.

A common property across the datasets is that each contains one label aligned with **semantic meaning** and another aligned with **pragmatics**. This reflects linguistic theories on the orthogonality of literal and expressive meaning (Potts, 2007).

3.2 Embedding Models

Since instruction prefixing requires an instruction-tuned model, we use **NV-Embed** (Lee et al., 2024) as the primary model for our experiments and analysis. Other instruction-tuned models, such as Multilingual E5 Large Instruct (Wang et al., 2024), exhibit similar behavior and tradeoffs across all datasets (Appendix Tables 13, 14).[1]

Unlike instruction prefixing, LLM preprocessing involves fully transforming the input text *before* the embedding stage. This means it can in principle be used with any embedding model, not just instruction-tuned ones. Therefore, for comparison we also run LLM preprocessing experiments with two non-instruction-tuned embedders: basic TF-IDF, and SBERT (Reimers and Gurevych, 2021).

[1]For space and readability, our analysis focuses solely on NV-Embed results since it was the top model on MTEB at the start of our experiments; results for other models can be found in the Appendix.

3.3 Prompt Strategies

We designed and tested several prompt strategies to inject perspective.

1. Baseline: No prefix instruction or preprocessing. The raw text is embedded directly.
2. Clustering Instruction Prefix: The text is prepended with a simple instruction defining the clustering goal. *Ex: Identify the topic of this customer support inquiry.*
3. Clustering Instruction Prefix with Classes: The prefix includes the explicit class names for the desired perspective. This tests the impact of providing more prior knowledge. *Ex: Identify the topic (financial, content, account, distribution) of this customer support inquiry.*
4. Inclusion LLM Preprocessing: A one-shot prompt instructs an LLM (GPT-4.1-mini) to rewrite the text, keeping only information relevant to the target perspective. *Ex: From this customer support inquiry, only keep text related to the topic. For example, "There's a bloody issue with my damn account" should become "There's an issue with my account."*
5. Exclusion LLM Preprocessing: A one-shot prompt instructs the LLM to rewrite the text, removing information related to a confounding perspective. *Ex: From this customer support inquiry, remove any text related to sentiment. For example, "There's a bloody annoying issue with my account" should become "There's an issue with my account."*

3.4 Clustering & Evaluation

We used **k-means** to cluster the generated embeddings, with the number of clusters k set to the number of ground-truth labels for the target perspective. Our methods operate at the embedding level and are therefore compatible with any clustering algorithm; k-means was chosen for its simplicity and widespread use. To account for variability, each experiment was repeated 10 times with different random seeds, and we report the mean scores.

We assessed clustering quality using the *V-Measure Score* (Rosenberg and Hirschberg, 2007), a standard metric from MTEB (Muennighoff et al., 2022) which is computed as the harmonic mean of homogeneity and completeness.[2]

[2]We also computed *Purity Score*, another standard MTEB metric; the results were qualitatively similar and are omitted for redundancy, but can be found in the Appendix.

To contextualize our results, we implement two reference points for comparison: Viswanathan et al. (2023)'s **Keyphrase Expansion** technique,[3] and a **Classification Prompt** where the LLM is asked to directly classify the text given the labels (e.g., *"Classify the topic (financial, content, account, distribution) of this customer support inquiry"*). The latter represents a theoretical upper bound for LLM performance in an unrealistic scenario with full prior knowledge of the classes.

4 Results & Analysis

4.1 Baselines

We first run clustering without prefix instructions or LLM preprocessing as a baseline. We find that across all settings, both SBERT and NV-Embed result in clusters that are closely aligned (high v-measure) with one perspective (which we denote as the *dominant perspective*, or DP) while being completely uncorrelated (near-zero v-measure) with the other perspective (which we denote as the *alternative perspective*, or AP) (See Figure 2 and Table 1)[4]. In all cases, the dominant perspective is the one associated with semantic meaning. In other words, the "default" behavior of the clustering pipeline appears to resemble topic modeling (Blei et al., 2003), corroborating prior work which has found that capturing intents or pragmatics requires deliberate finessing of the model (Zhang et al., 2017b).

Dataset	DP	AP
Customer Support	Inquiry Topic	Sentiment
StackOverflow	Programming Language	Question Intent
Coarse Discourse	Subreddit	Speech Type

Table 1: Dominant (DP) and Alternative (AP) perspectives for each dataset.

This finding illuminates a clear direction towards answering our RQs (Section 1): given that the baseline yields clusters that are closely correlated with only the dominant perspective, in order to demonstrate controllability of clusters, we should aim to shape the embeddings such that the resulting clusters are more correlated with the alternative perspective. Further analysis is geared towards achieving this goal.

[3]Specifically, a reimplementation using NV-Embed as the LLM to ensure a fair comparison.

[4]TF-IDF shows similar behavior except on the Reddit dataset, where it performs poorly across the board.

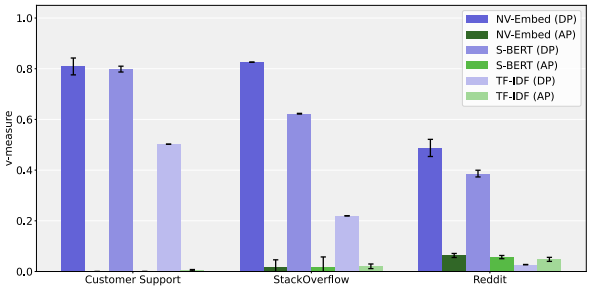

Figure 2: Baseline V-measure scores for Customer Support, StackOverflow, and Reddit datasets ("DP" refers to the dominant perspective, "AP" refers to the alternative perspective).

4.2 Perspective Injection Results

Using the four non-baseline prompting strategies described in Section 3.3, we develop prompts to steer embeddings towards the alternative perspective for each dataset,[5] then run k-means clustering on the resulting embeddings. For comparison, we also run k-means clustering on the baseline, no-prompt embeddings. The full results are summarized in Table 2, which shows v-measure scores for each clustering, alongside the baseline and reference methods for comparison. The following subsections discuss key findings in further detail.

Figure 3: PCA visualizations of NV-Embed embeddings on Customer Support dataset. Cluster Labels (left) and True Labels (right) are shown for baseline embeddings (top) versus using a prefix to guide the embeddings to favor the AP (bottom).

Prefixing improves alignment with the alternative perspective. Adding a prefix instruction for the AP generally results in clusters that are more

[5]The full text of the prompts is ommitted for space but can be found in Appendix Tables 15-18.

Prompt Strategy	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	Topic (DP)	Sentiment (AP)	Language (DP)	Intent (AP)	Subreddit (DP)	Speech (AP)
Baseline	0.81 ± 0.03	0.00 ± 0.00	0.83 ± 0.01	0.02 ± 0.00	0.49 ± 0.00	0.06 ± 0.00
Prefix AP	**0.04** ± 0.00	**0.73** ± 0.00	**0.72** ± 0.00	**0.04** ± 0.00	**0.06** ± 0.01	**0.22** ± 0.00
Prefix AP (w/ classes)	0.04 ± 0.00	0.95 ± 0.00	0.69 ± 0.01	0.04 ± 0.00	0.06 ± 0.00	0.20 ± 0.00
Inclusion AP	0.01 ± 0.00	0.98 ± 0.00	0.83 ± 0.01	0.02 ± 0.00	0.08 ± 0.01	0.13 ± 0.00
Exclusion DP	**0.05** ± 0.05	**0.70** ± 0.00	**0.27** ± 0.01	**0.29** ± 0.01	**0.27** ± 0.00	**0.12** ± 0.00
Keyphrase (DP)	0.70 ± —	0.00 ± —	0.83 ± —	0.02 ± —	0.42 ± —	0.06 ± —
Keyphrase (AP)	0.00 ± —	0.82 ± —	0.64 ± —	0.016 ± —	0.19 ± —	0.10 ± —
Classify	0.84 ± 0.00	0.92 ± 0.00	0.81 ± 0.08	0.46 ± 0.00	0.54 ± 0.00	0.22 ± 0.00

Table 2: V-measure scores were calculated to evaluate clustering performance for each prompt method on three datasets. The Prefix AP and Exclusion DP methods (bolded) consistently improved v-measure scores for the alternative perspective AP for all three datasets. LLM Few Shot results use the NV-Embed2 embedder. Maximum values among methods where explicit labels are unavailable (ie, all but Classify and Prefix AP (w/ classes)) are underlined. ± terms indicate standard deviations. Refer to Table 1 for AP and DP definitions.

closely correlated with that perspective. This effect is most significant for the Customer Support dataset: baseline v-measure for the AP (sentiment) was 0 (equivalent to random performance), but the prefix-induced clusters had 0.73 v-measure (Table 2). This effect is also visually demonstrated in Figure 3, which shows how prefixing altered the geometry of the embedding space to more cleanly separate positive-sentiment from negative-sentiment examples.

The results for the Reddit dataset are more muted, but still show a noticeable improvement: the v-measure for the AP (speech type) is 0.22 after prefixing, versus 0.06 baseline. The least improvement is in the StackOverflow dataset, where even with prefixing, the v-measure for the AP (intent) remains low, at 0.04 (versus 0.02 baseline). This may be because properties related to programming language (DP) are ubiquitous in StackOverflow posts, and intent detection is inherently a more difficult task (Sultana et al., 2024; Sanchez-Karhunen et al., 2024).

Prefixing does not require in-depth knowledge of specific classes. The prefixing results suggest a positive answer to RQ1: it is possible to steer embeddings towards a desired, underrepresented perspective (AP) using a natural language prefix prompt. Next, we investigate RQ2: does more effective control over the clusters require more a priori knowledge about the data? To test this, we contrast the previous results, which used a generic instruction to focus on the desired perspective, with a more *unrealistic* variant where the prompt includes the specific clusters (labels) to look for (prompt strategy 3 in Section 3.3).

The results are shown in Table 2, row 3. Overall, we find that listing explicit classes does not generally improve performance; it has effectively no impact on the v-measures for the StackOverflow and Reddit datasets, with only the Customer Support dataset showing a noticeable improvement (from 0.73 to 0.95).

These results suggest that the effectiveness of prefixing does not depend on unrealistic prior knowledge of what the clusters should be—a positive indicator that this strategy has practical utility in real-world applications, where clustering is often used for discovering previously unknown properties of the data.

LLM Preprocessing: Exclusion improves the alternative perspective performance across the board, while inclusion is less consistent. As discussed in Section 3.3, we consider two ways of formulating the LLM preprocessing prompt: inclusion and exclusion. In the specific context of steering the embeddings towards AP, the inclusion prompt takes the form of instructing the LLM to focus on the AP (ex, "only keep text related to sentiment"), while exclusion involves instructing the LLM to ignore the dominant perspective (ex, "remove any text related to the inquiry topic").

The v-measure results when using inclusion preprocessing (Table 2, row 4) are qualitatively similar to the results from prefixing: a large improvement on AP for Customer Support (0.94) and a modest improvement for Reddit (0.20), but no improvement for StackOverflow. By contrast, exclusion preprocessing (row 5) more consistently results in improvements across all settings: 0.70 for Customer Support, 0.12 for Reddit, and 0.29

Prompt Strategy	TF-IDF	SBERT
Baseline	0.01 ± 0.01	0.00 ± 0.00
Prefix AP	0.01 ± 0.01	0.00 ± 0.00
Prefix AP (w/ classes)	0.01 ± 0.01	0.00 ± 0.00
Inclusion AP	0.20 ± 0.00	0.58 ± 0.00
Exclusion DP	0.16 ± 0.01	0.46 ± 0.00

Table 3: v-measure scores for the AP on the Customer Support dataset, using TF-IDF (left) and SBERT (right) as the embedding model (instead of NV-Embed).

for StackOverflow.

Notably, exclusion preprocessing is the only strategy that achieves a nontrivial improvement in AP v-measure for StackOverflow. This result suggests a key advantage of the exclusion strategy: by explicitly excluding the DP, it can retain effectiveness even in settings where the DP may be especially prominent (as in StackOverflow, where posts are inherently always about programming).

Prefixing is incompatible with non-instruction-tuned embedders, but LLM preprocessing can still be effective. As noted in Section 3.2, instruction prefixing requires an instruction-tuned model whereas LLM preprocessing is theoretically compatible with any embedding model. To quantify this difference, we conduct a follow-up experiment with our two non-instruction-tuned models, TF-IDF and SBERT.

As expected, when prefixing is used with TF-IDF or SBERT, it has no effect (Table 3, Rows 2-3)—an unsurprising finding, given that such models were not specifically trained to accept instructions, and would therefore interpret the instruction prefix as just regular text.

By contrast, inclusion and exclusion preprocessing on the Customer Support dataset still leads to improved v-measure scores for the AP even when the embedding model is SBERT or, more impressively, something as basic as TF-IDF (Table 3, Rows 4-5). This effect appears to be limited to the Customer Service dataset, however: repeating this experiment on the StackOverflow and Reddit datasets yielded only modest to negligible gains in v-measure.[6]

These mixed results highlight key subtleties in the comparison between instruction prefixing and LLM preprocessing, and between instruction-tuned and non-instruction-tuned models. Instruction-tuned models still show the best results across all datasets for both perspective injection techniques, indicating that the advanced language understanding capabilities of modern LLM-based embedders are still required to unlock these techniques' full potential. Nonetheless, the Customer Support results suggest that for less complex datasets, simpler non-instruction-tuned models like SBERT may still be responsive to LLM preprocessing. This finding may have practical benefits, as non-instruction-tuned models are less resource intensive and can therefore be preferred in some contexts. Further work is needed to more fully characterize the circumstances under which non-instruction-tuned models are a viable option for use with LLM preprocessing.

Perspective injection requires tradeoffs. Our results show that for a given dataset, improving performance for the AP comes at the expense of significantly decreasing performance for the DP, and vice versa. As seen in Table 2 row 1, across all three datasets, baseline embeddings have high v-measure scores and thus high performance for the DP. The same embeddings perform poorly for the AP, with v-measure scores near 0.

When we use Prefix AP to improve the performance for the AP (Table 2 row 2), we observe a sharp decline in performance of these embeddings for the DP. This tradeoff in performance is consistently noted for the Inclusion AP and Exclusion DP methods, which both improve the embeddings' performance for the AP. This 'tradeoff' is also somewhat proportional, as seen in Table 2. When perspective injection greatly increases performance for the AP (such as with the Customer Support dataset), then the performance for the DP using the same embeddings tuned to the AP drops to near zero or random selection. When the AP injection performs poorly with small improvements (such as prefixing and inclusion preprocessing for the StackOverflow dataset), we observe that the DP performance with those embeddings still remains relatively high.

5 Conclusion

In this paper, we explore user-friendly methods to control the perspectives favored by embeddings for unsupervised clustering. We find that baseline embeddings tend to favor a dominant perspective for a given dataset, but it is possible to control the embeddings to favor the alternative perspective through lightweight human input, such as instruc-

[6]Exact values can be found in Appendix Tables 10 and 12.

tion prefixing or LLM preprocessing. Prefixed instructions are effective only for instruction-tuned LLM-based embedding models. LLM preprocessing can be effective on traditional lightweight BERT embedding models when leveraged on simple text data, but requires prompting a language model once per text entry (i.e., per dataset row). This highlights the unique advantages of LLM-derived embedding models over traditional counterparts.

Relying only on zero- and one-shot prompts, the methods introduced in this paper are particularly well-suited for early-stage exploratory text analysis, particularly when there is little prior knowledge and/or when examining data from multiple distinct perspectives is useful. This is particularly relevant in our domain of interest (enterprise AI and specifically customer support), where Process Owners may know which dimensions to prioritize (*inclusion*, e.g., customer-reported symptom or severity), or know which aspects are irrelevant to their use case (*exclusion*, e.g. programming language or system). Our methods enable these preferences to be reflected in the resulting clusters.

Limitations

Our approaches, while effective, have several limitations. First, we evaluate them on a limited number of reproducible, public datasets described in the paper. We also test a limited number of LLMs, and we do not conduct an in-depth analysis on how the observed properties vary across LLM type or size. Our analysis focuses on dominant and alternative perspectives, and we do not explore the scenario where there are potentially a higher number of dimensions of interest. In enterprise datasets, there are often more than two relevant dimensions, and we have not explored how the strategies presented here generalize to such multiple perspectives. Finally, further work is needed to validate the results of exclusion preprocessing, by conducting tests in other domains with similarly prominent dominant perspectives.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Bitext. 2024. Bitext gen ai chatbot customer support dataset.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Alexander Butyaev, Chrisostomos Drogaris, Olivier Tremblay-Savard, and Jérôme Waldispühl. 2022. Human-supervised clustering of multidimensional data using crowdsourcing. *Royal Society open science*, 9(5):211189.

Hongliu Cao. 2024. Recent advances in text embedding: A comprehensive review of top-performing methods on the mteb benchmark. *ArXiv*, abs/2406.01607.

Rich Caruana. 2013. Clustering: Probably approximately useless? In *CIKM*, volume 13, pages 1259–1260. Citeseer.

Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. 2006. Meta clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 107–118. IEEE.

Garrett J Chan, Mark Fung, Jill Warrington, and Sarah A Nowak. 2025. Understanding health-related discussions on reddit: Development of a topic assignment method and exploratory analysis. *JMIR Formative Research*, 9(1):e55309.

Yuan Chang, Ziyue Li, and Xiaoqiu Le. 2024. Guiding large language models via external attention prompting for scientific extreme summarization. In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 226–242.

Sajib Dasgupta, Richard Golden, and Vincent Ng. 2012. Clustering documents along multiple dimensions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 879–885.

Yilun Hua, Nicholas Chernogor, Yuzhe Gu, Seoyeon Julie Jeong, Miranda Luo, and Cristian Danescu-Niculescu-Mizil. 2024. How did we get here? summarizing conversation dynamics. *arXiv preprint arXiv:2404.19007*.

Deepak Kumar, Jeff Hancock, Kurt Thomas, and Zakir Durumeric. 2023. Understanding the behaviors of toxic accounts on reddit. *Proceedings of the ACM Web Conference 2023*.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023. Guiding large language models via directional stimulus prompting. *Advances in Neural Information Processing Systems*, 36:62630–62656.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Zhijie Nie, Zhangchi Feng, Mingxin Li, Cunwang Zhang, Yanzhao Zhang, Dingkun Long, and Richong Zhang. 2024. When text embedding meets large language model: A comprehensive survey. *ArXiv*, abs/2412.09165.

OpenAI. 2025. OpenAI API Pricing. https://openai.com/api/pricing/. Accessed: 2025-10-01.

Haniyeh Ehsani Oskouie, Christina Chance, Claire Huang, Margaret Capetz, Elizabeth Eyeson, and Majid Sarrafzadeh. 2024. Leveraging large language models and topic modeling for toxicity classification. *arXiv preprint arXiv:2411.17876*.

Alina Petukhova, João P Matos-Carvalho, and Nuno Fachada. 2025. Text clustering with large language model embeddings. *International Journal of Cognitive Computing in Engineering*, 6:100–108.

Christopher Potts. 2007. The Expressive Dimension. *Theoretical Linguistics*, 33(2):165–198.

Yue Qin, Shifei Ding, Lijuan Wang, and Yanru Wang. 2019. Research progress on semi-supervised clustering. *Cognitive Computation*, 11(5):599–612.

Nils Reimers and Iryna Gurevych. 2021. all-minilm-l6-v2. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2. Accessed: 2025-04-27.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Eduardo Sanchez-Karhunen, Jose F Quesada-Moreno, and Miguel A Gutiérrez-Naranjo. 2024. Interpretation of the intent detection problem as dynamics in a low-dimensional space. *arXiv preprint arXiv:2408.02838*.

Nidhish Shah, Zulkuf Genc, and Dogu Araci. 2024. Stackeval: Benchmarking llms in coding assistance. *Advances in Neural Information Processing Systems*, 37:36976–36994.

Tangina Sultana, Ashis Kumar Mandal, Hasi Saha, Md Nahid Sultan, and Md Delowar Hossain. 2024. Intent identification by semantically analyzing the search query. *Modelling*, 5(1):292–314.

Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Zhengwei Tao, and Shuai Ma. 2024. Llms are also effective embedding models: An in-depth overview. *arXiv preprint arXiv:2412.12591*.

Puja Trivedi, Nurendra Choudhary, E-Wen Huang, Karthik Subbian, and Danai Koutra. Large language model guided graph clustering.

Vijay Viswanathan, Kiril Gashteovski, Carolin (Haas) Lawrence, Tongshuang Sherry Wu, and Graham Neubig. 2023. Large language models enable few-shot clustering. *Transactions of the Association for Computational Linguistics*, 12:321–333.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Chen Yang, Bin Cao, and Jing Fan. 2024. Tec: A novel method for text clustering with large language models guidance and weakly-supervised contrastive learning. In *International Conference on Web and Social Media*.

Zhou Yang, Spencer D. Bradshaw, Rattikorn Hewett, and Fang Jin. 2019. Discovering opioid use patterns from social media for relapse prevention. *Computer*, 55:23–33.

Amy Zhang, Bryan Culbertson, and Praveen Paritosh. 2017a. Characterizing online discussion using coarse discourse sequences. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):357–366.

Justine Zhang, Arthur Spirling, and Cristian Danescu-Niculescu-Mizil. 2017b. Asking too Much? The Rhetorical Role of Questions in Political Discourse. In *Proceedings of EMNLP*.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. Clusterllm: Large language models as a guide for text clustering. *ArXiv*, abs/2305.14871.

6 Appendix

Cost Type	Price (per 1M tokens)
Input	$0.80
Cached Input	$0.20
Output	$3.20

Table 4: Costs for GPT-4.1-mini (OpenAI, 2025).

6.1 Keyphrase Expansion Results

Tables 5 and 6 compare the performance of two different embedding models using few-shot and zero-shot clustering with keyphrase expansion. Below, you can find the prompts used for the experiments in keyphrase expansion.

6.1.1 Additional Prompts for Keyphrase Expansion

Customer Service Dataset:

- Topic: I am trying to cluster customer service queries based on whether they fall into the same topic. To help me with this, for a given user query, provide a comprehensive set of keyphrases that could describe this query's topic. These keyphrases should be distinct from those that might describe queries with different topics. Generate the set of keyphrases as a JSON-formatted list.

  Query: "There's an issue with my account"

  Keyphrases: ["account", "issue"]

- Sentiment: I am trying to cluster customer service queries based on whether they express the same general user sentiment. To help me with this, for a given user query, provide a comprehensive set of keyphrases that could describe this query's sentiment. These keyphrases should be distinct from those that might describe queries with different intents. Generate the set of keyphrases as a JSON-formatted list.

  Query: "I love this product"

  Keyphrases: ["positive", "love"]"""

StackOverflow Dataset:

- Programming Languge: I am trying to cluster StackOverflow posts based on whether they use the same programming language. To help me with this, for a given user query, provide a comprehensive set of keyphrases that could describe this query's programming language. These keyphrases should be distinct from those that might describe queries with different languages. Generate the set of keyphrases as a JSON-formatted list.

  Query: "I'm having issues with this python enum"

  Keyphrases: ["python", "enum"]

- Question Type: I am trying to cluster StackOverflow posts based on whether they have the same question type/intent. To help me with this, for a given user query, provide a comprehensive set of keyphrases that could describe this query's question type. These keyphrases should be distinct from those that might describe queries with different question types. Generate the set of keyphrases as a JSON-formatted list.

  Query: "I don't know why this code isn't working"

  Keyphrases: ["debugging", "confusion"]

Reddit Dataset:

- Subreddit: I am trying to cluster Reddit posts based on whether they are from the same subreddit. To help me with this, for a given user query, provide a comprehensive set of keyphrases that could describe this query's subreddit. These keyphrases should be distinct from those that might describe queries with different subreddits. Generate the set of keyphrases as a JSON-formatted list.

  Query: "Let me clarify this point about ahri from league of legends"

  Keyphrases: ["league of legends", "ahri"]

- Speech Act: I am trying to cluster Reddit posts based on whether they have the same discourse type. To help me with this, for a given user query, provide a comprehensive set of keyphrases that could describe this query's discourse type. These keyphrases should be distinct from those that might describe queries with different discourse types. Generate the set of keyphrases as a JSON-formatted list.

  Query: "Let me clarify this point about ahri from league of legends"

  Keyphrases: ["clarify", "point"]

Method / Metric	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
DP – v-measure	.995	0.00	0.704	0.046	0.418	0.051
AP – v-measure	0.00	0.801	0.583	0.032	0.433	0.045
DP – purity	.999	0.504	0.871	0.556	0.792	0.515
AP – purity	0.280	0.969	0.679	0.534	0.748	0.515

Table 5: LLM Few Shot Clustering keyphrase expansion technique (InstructOR-xl). DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method / Metric	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
DP – v-measure	0.6989	0.0002	0.8265	0.0162	0.4191	0.0624
AP – v-measure	0.0004	0.8166	0.6359	0.0160	0.1876	0.0988
DP – purity	0.724	0.510	0.9324	0.5242	0.7857	0.5150
AP – purity	0.279	0.972	0.7101	0.5193	0.5679	0.5167

Table 6: LLM Few Shot Clustering keyphrase expansion technique (NV-Embed). DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

6.2 Detailed Results

Tables 7 and 8 provide more granular details about the NV-Embed results in Table 1. Moreover, the detailed results of injecting perspective with SBERT and TF-IDF embeddings are provided in Tables 9, 10, 11, and 12.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.85 ± 0.02	0.50 ± 0.00	0.93 ± 0.01	0.53 ± 0.00	0.78 ± 0.00	0.52 ± 0.00
Prefix DP	0.82 ± 0.00	0.50 ± 0.00	0.93 ± 0.00	0.53 ± 0.00	0.79 ± 0.00	0.52 ± 0.00
Prefix DP (with classes)	0.82 ± 0.00	0.50 ± 0.00	0.93 ± 0.00	0.53 ± 0.00	0.80 ± 0.00	0.52 ± 0.00
Prefix AP	0.36 ± 0.00	0.95 ± 0.00	0.71 ± 0.00	0.53 ± 0.00	0.49 ± 0.00	0.65 ± 0.00
Prefix AP (with classes)	0.35 ± 0.00	1.00 ± 0.00	0.70 ± 0.00	0.53 ± 0.00	0.49 ± 0.00	0.65 ± 0.00
Inclusion DP	0.86 ± 0.01	0.50 ± 0.00	0.94 ± 0.00	0.53 ± 0.00	0.67 ± 0.00	0.55 ± 0.01
Exclusion AP	0.84 ± 0.02	0.50 ± 0.00	0.94 ± 0.00	0.53 ± 0.00	0.77 ± 0.00	0.52 ± 0.00
Inclusion AP	0.29 ± 0.00	1.00 ± 0.00	0.94 ± 0.00	0.53 ± 0.00	0.51 ± 0.01	0.57 ± 0.00
Exclusion DP	0.34 ± 0.03	0.95 ± 0.00	0.56 ± 0.01	0.74 ± 0.00	0.67 ± 0.00	0.57 ± 0.00
Prefix + Inclusion	0.82 ± 0.00	0.94 ± 0.00	0.94 ± 0.00	0.52 ± 0.00	0.78 ± 0.00	0.62 ± 0.00
Classify	0.94 ± 0.00	0.99 ± 0.00	0.89 ± 0.00	0.81 ± 0.00	0.81 ± 0.00	0.64 ± 0.00

Table 7: NV-Embed purity scores over all prompting strategies. DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.81 ± 0.03	0.00 ± 0.00	0.83 ± 0.01	0.02 ± 0.00	0.49 ± 0.00	0.06 ± 0.00
Prefix DP	0.78 ± 0.00	0.00 ± 0.00	0.83 ± 0.01	0.02 ± 0.00	0.52 ± 0.00	0.07 ± 0.00
Prefix DP (with classes)	0.78 ± 0.00	0.00 ± 0.00	0.84 ± 0.00	0.02 ± 0.00	0.54 ± 0.00	0.07 ± 0.00
Prefix AP	0.04 ± 0.00	0.73 ± 0.00	0.72 ± 0.00	0.04 ± 0.00	0.06 ± 0.01	0.22 ± 0.00
Prefix AP (with classes)	0.04 ± 0.00	0.95 ± 0.00	0.69 ± 0.01	0.04 ± 0.00	0.06 ± 0.00	0.20 ± 0.00
Inclusion DP	0.82 ± 0.00	0.00 ± 0.00	0.83 ± 0.01	0.02 ± 0.00	0.30 ± 0.00	0.09 ± 0.01
Exclusion AP	0.81 ± 0.02	0.00 ± 0.00	0.84 ± 0.01	0.02 ± 0.00	0.45 ± 0.00	0.06 ± 0.00
Inclusion AP	0.01 ± 0.00	0.98 ± 0.00	0.83 ± 0.01	0.02 ± 0.00	0.08 ± 0.01	0.13 ± 0.00
Exclusion DP	0.05 ± 0.05	0.70 ± 0.00	0.27 ± 0.01	0.29 ± 0.01	0.27 ± 0.00	0.12 ± 0.00
Prefix + Inclusion	0.79 ± 0.00	0.71 ± 0.00	0.85 ± 0.00	0.02 ± 0.00	0.50 ± 0.00	0.16 ± 0.00
Classify	0.84 ± 0.00	0.92 ± 0.00	0.81 ± 0.08	0.46 ± 0.00	0.54 ± 0.00	0.22 ± 0.00

Table 8: NV-Embed v-measure scores over all prompting strategies. DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.89 ± 0.05	0.50 ± 0.00	0.76 ± 0.02	0.53 ± 0.00	0.72 ± 0.00	0.52 ± 0.00
Prefix DP	0.87 ± 0.04	0.50 ± 0.00	0.75 ± 0.01	0.53 ± 0.00	0.67 ± 0.00	0.52 ± 0.00
Prefix DP (with classes)	0.76 ± 0.00	0.50 ± 0.00	0.70 ± 0.01	0.56 ± 0.01	0.66 ± 0.00	0.52 ± 0.00
Prefix AP	0.71 ± 0.00	0.51 ± 0.00	0.77 ± 0.01	0.53 ± 0.00	0.66 ± 0.00	0.52 ± 0.00
Prefix AP (with classes)	0.87 ± 0.01	0.51 ± 0.00	0.77 ± 0.01	0.53 ± 0.00	0.61 ± 0.00	0.52 ± 0.00
Inclusion DP	0.95 ± 0.04	0.51 ± 0.00	0.93 ± 0.00	0.53 ± 0.00	0.68 ± 0.00	0.52 ± 0.00
Exclusion AP	0.96 ± 0.01	0.51 ± 0.00	0.92 ± 0.01	0.53 ± 0.00	0.69 ± 0.00	0.52 ± 0.00
Inclusion AP	0.31 ± 0.00	0.88 ± 0.00	0.90 ± 0.01	0.57 ± 0.00	0.69 ± 0.00	0.52 ± 0.00
Exclusion DP	0.32 ± 0.04	0.82 ± 0.05	0.51 ± 0.03	0.60 ± 0.01	0.69 ± 0.00	0.52 ± 0.00
Prefix + Inclusion	0.90 ± 0.00	1.00 ± 0.00	0.92 ± 0.00	0.55 ± 0.00	0.69 ± 0.00	0.52 ± 0.00
Classify	0.94 ± 0.00	0.98 ± 0.00	0.89 ± 0.00	0.81 ± 0.00	0.81 ± 0.00	0.64 ± 0.00

Table 9: SBERT purity scores. Prefixing and preprocessing prompts are identical to those used in NVEmbed, DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.80 ± 0.03	0.00 ± 0.00	0.62 ± 0.01	0.02 ± 0.00	0.39 ± 0.00	0.06 ± 0.00
Prefix DP	0.77 ± 0.05	0.00 ± 0.00	0.62 ± 0.01	0.02 ± 0.00	0.29 ± 0.00	0.05 ± 0.00
Prefix DP (with classes)	0.64 ± 0.00	0.00 ± 0.00	0.62 ± 0.01	0.03 ± 0.01	0.20 ± 0.00	0.05 ± 0.05
Prefix AP	0.61 ± 0.00	0.00 ± 0.00	0.63 ± 0.01	0.02 ± 0.00	0.29 ± 0.00	0.05 ± 0.00
Prefix AP (with classes)	0.77 ± 0.01	0.00 ± 0.00	0.63 ± 0.01	0.02 ± 0.00	0.22 ± 0.00	0.05 ± 0.00
Inclusion DP	0.90 ± 0.04	0.00 ± 0.00	0.81 ± 0.01	0.02 ± 0.00	0.31 ± 0.01	0.05 ± 0.00
Exclusion AP	0.90 ± 0.03	0.00 ± 0.00	0.78 ± 0.01	0.02 ± 0.00	0.39 ± 0.00	0.05 ± 0.00
Inclusion AP	0.02 ± 0.00	0.58 ± 0.00	0.71 ± 0.01	0.04 ± 0.00	0.37 ± 0.00	0.05 ± 0.00
Exclusion DP	0.02 ± 0.03	0.46 ± 0.10	0.20 ± 0.03	0.06 ± 0.01	0.38 ± 0.00	0.05 ± 0.00
Prefix + Inclusion	0.81 ± 0.00	0.97 ± 0.00	0.79 ± 0.01	0.03 ± 0.00	0.39 ± 0.00	0.04 ± 0.00
Classify	0.84 ± 0.00	0.86 ± 0.00	0.81 ± 0.08	0.46 ± 0.00	0.54 ± 0.00	0.22 ± 0.00

Table 10: SBERT v-measure scores. Prefixing and preprocessing prompts are identical to those used in NV-Embed, DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.69 ± 0.04	0.52 ± 0.02	0.48 ± 0.03	0.51 ± 0.02	0.49 ± 0.01	0.54 ± 0.01
Prefix DP	0.57 ± 0.06	0.52 ± 0.02	0.47 ± 0.03	0.52 ± 0.02	0.49 ± 0.00	0.53 ± 0.01
Prefix DP (with classes)	0.44 ± 0.08	0.56 ± 0.05	0.47 ± 0.04	0.52 ± 0.03	0.49 ± 0.00	0.53 ± 0.01
Prefix AP	0.57 ± 0.06	0.52 ± 0.02	0.45 ± 0.04	0.51 ± 0.04	0.49 ± 0.00	0.53 ± 0.01
Prefix AP (with classes)	0.56 ± 0.05	0.53 ± 0.02	0.44 ± 0.03	0.52 ± 0.03	0.49 ± 0.00	0.53 ± 0.01
Inclusion DP	0.67 ± 0.08	0.59 ± 0.06	0.62 ± 0.09	0.51 ± 0.02	0.48 ± 0.00	0.53 ± 0.01
Exclusion AP	0.71 ± 0.00	0.52 ± 0.02	0.52 ± 0.04	0.51 ± 0.02	0.48 ± 0.00	0.53 ± 0.00
Inclusion AP	0.29 ± 0.00	0.65 ± 0.00	0.51 ± 0.09	0.60 ± 0.05	0.48 ± 0.00	0.52 ± 0.00
Exclusion DP	0.30 ± 0.00	0.61 ± 0.01	0.50 ± 0.03	0.68 ± 0.05	0.48 ± 0.00	0.52 ± 0.01
Prefix + Inclusion	0.61 ± 0.05	0.63 ± 0.00	0.60 ± 0.06	0.55 ± 0.03	0.48 ± 0.00	0.52 ± 0.01
Classify	0.94 ± 0.00	0.98 ± 0.00	0.89 ± 0.00	0.81 ± 0.00	0.81 ± 0.00	0.63 ± 0.00

Table 11: TF-IDF purity scores. Prefixing and preprocessing prompts are identical to those used in NV-Embed, DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.50 ± 0.03	0.01 ± 0.01	0.22 ± 0.04	0.02 ± 0.01	0.03 ± 0.01	0.05 ± 0.01
Prefix DP	0.35 ± 0.08	0.01 ± 0.01	0.21 ± 0.03	0.03 ± 0.01	0.03 ± 0.00	0.05 ± 0.01
Prefix DP (with classes)	0.17 ± 0.12	0.03 ± 0.03	0.20 ± 0.04	0.03 ± 0.01	0.03 ± 0.00	0.04 ± 0.01
Prefix AP	0.35 ± 0.08	0.01 ± 0.01	0.18 ± 0.04	0.02 ± 0.02	0.03 ± 0.00	0.05 ± 0.01
Prefix AP (with classes)	0.37 ± 0.06	0.01 ± 0.01	0.18 ± 0.04	0.02 ± 0.02	0.03 ± 0.00	0.0 ± 0.01
Inclusion DP	0.47 ± 0.11	0.04 ± 0.03	0.38 ± 0.08	0.02 ± 0.00	0.00 ± 0.00	0.05 ± 0.00
Exclusion AP	0.52 ± 0.01	0.01 ± 0.01	0.26 ± 0.03	0.02 ± 0.01	0.02 ± 0.01	0.05 ± 0.01
Inclusion AP	0.00 ± 0.00	0.20 ± 0.00	0.23 ± 0.07	0.07 ± 0.03	0.01 ± 0.00	0.02 ± 0.00
Exclusion DP	0.01 ± 0.00	0.16 ± 0.01	0.19 ± 0.03	0.25 ± 0.04	0.02 ± 0.01	0.04 ± 0.01
Prefix + Inclusion	0.37 ± 0.06	0.18 ± 0.00	0.35 ± 0.07	0.04 ± 0.02	0.01 ± 0.00	0.05 ± 0.01
Classify	0.84 ± 0.00	0.86 ± 0.00	0.81 ± 0.08	0.46 ± 0.00	0.54 ± 0.00	0.22 ± 0.00

Table 12: TF-IDF v-measure scores. Prefixing and preprocessing prompts are identical to those used in NV-Embed, DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	0.98 ± 0.00	0.00 ± 0.00	0.65 ± 0.01	0.02 ± 0.00	0.45 ± 0.00	0.06 ± 0.00
Prefix DP	0.99 ± 0.00	0.00 ± 0.00	0.75 ± 0.00	0.02 ± 0.00	0.41 ± 0.00	0.05 ± 0.00
Prefix DP (with classes)	0.89 ± 0.00	0.00 ± 0.00	0.81 ± 0.00	0.02 ± 0.00	0.42 ± 0.00	0.06 ± 0.00
Prefix AP	0.36 ± 0.00	0.34 ± 0.00	0.72 ± 0.03	0.02 ± 0.00	0.30 ± 0.00	0.12 ± 0.00
Prefix AP (with classes)	0.09 ± 0.00	0.34 ± 0.00	0.58 ± 0.02	0.02 ± 0.00	0.09 ± 0.00	0.11 ± 0.00
Inclusion DP	0.97 ± 0.00	0.00 ± 0.00	0.68 ± 0.02	0.02 ± 0.00	0.27 ± 0.00	0.04 ± 0.00
Exclusion AP	0.97 ± 0.00	0.00 ± 0.00	0.77 ± 0.01	0.02 ± 0.00	0.42 ± 0.00	0.05 ± 0.00
Inclusion AP	0.02 ± 0.00	0.95 ± 0.00	0.68 ± 0.02	0.01 ± 0.00	0.22 ± 0.00	0.12 ± 0.00
Exclusion DP	0.00 ± 0.00	0.67 ± 0.00	0.12 ± 0.02	0.37 ± 0.01	0.25 ± 0.00	0.12 ± 0.00
Prefix + Inclusion	1.00 ± 0.00	0.91 ± 0.00	0.69 ± 0.01	0.02 ± 0.00	0.05 ± 0.00	0.15 ± 0.00
Classify	0.84 ± 0.00	0.86 ± 0.00	0.81 ± 0.00	0.46 ± 0.00	0.54 ± 0.00	0.22 ± 0.00

Table 13: Multilingual-E5-large-instruct (Wang et al., 2024) v-measure scores. Prefixing and preprocessing prompts are identical to those used in NV-Embed, DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Customer Support Inquiries		StackOverflow Questions		Reddit Posts	
	DP	AP	DP	AP	DP	AP
Baseline	1.00 ± 0.00	0.50 ± 0.00	0.73 ± 0.01	0.52 ± 0.00	0.77 ± 0.00	0.52 ± 0.00
Prefix DP	1.00 ± 0.00	0.50 ± 0.00	0.90 ± 0.00	0.53 ± 0.00	0.74 ± 0.00	0.52 ± 0.00
Prefix DP (with classes)	0.97 ± 0.00	0.51 ± 0.00	0.93 ± 0.00	0.52 ± 0.00	0.73 ± 0.00	0.52 ± 0.00
Prefix AP	0.59 ± 0.00	0.83 ± 0.00	0.87 ± 0.02	0.52 ± 0.00	0.68 ± 0.00	0.58 ± 0.00
Prefix AP (with classes)	0.38 ± 0.00	0.83 ± 0.00	0.66 ± 0.01	0.52 ± 0.00	0.51 ± 0.00	0.57 ± 0.00
Inclusion DP	0.99 ± 0.00	0.50 ± 0.00	0.81 ± 0.03	0.53 ± 0.00	0.68 ± 0.00	0.52 ± 0.00
Exclusion AP	0.99 ± 0.00	0.50 ± 0.00	0.91 ± 0.00	0.53 ± 0.00	0.76 ± 0.00	0.52 ± 0.00
Inclusion AP	0.31 ± 0.00	0.99 ± 0.00	0.81 ± 0.03	0.52 ± 0.00	0.64 ± 0.00	0.57 ± 0.00
Exclusion DP	0.29 ± 0.00	0.93 ± 0.00	0.44 ± 0.02	0.76 ± 0.00	0.66 ± 0.00	0.58 ± 0.00
Prefix + Inclusion	1.00 ± 0.00	0.99 ± 0.00	0.86 ± 0.01	0.52 ± 0.00	0.48 ± 0.00	0.61 ± 0.00
Classify	0.94 ± 0.00	0.98 ± 0.00	0.89 ± 0.00	0.81 ± 0.00	0.81 ± 0.00	0.64 ± 0.00

Table 14: Multilingual-E5-large-instruct (Wang et al., 2024) purity scores. Prefixing and preprocessing prompts are identical to those used in NV-Embed, DP and AP correspond to dominant and alternative perspectives per dataset, specified in Table 1.

Method	Prefix	LLM Preprocessing Prompt
Baseline	-	-
Prefix DP	Identify the topic of this customer support inquiry.	-
Prefix DP (with classes)	Identify the topic (financial, content, account, distribution) of this customer support inquiry.	-
Prefix AP	Identify the sentiment of this customer support inquiry.	-
Prefix AP (with classes)	Identify the sentiment (rude or polite) of this customer support inquiry.	-
Inclusion DP	-	From this customer support inquiry, only keep text related to the topic. For example, "There's a bloody issue with my damn account" should become "There's an issue with my account."
Inclusion + prefix DP	Identify the topic of this customer support inquiry.	From this customer support inquiry, only keep text related to the topic. For example, "There's a bloody issue with my damn account" should become "There's an issue with my account."
Exclusion AP	-	From this customer support inquiry, remove any text related to sentiment. For example, "There's a bloody issue with my damn account" should become "There's an issue with my account."
Inclusion AP	-	From this customer support inquiry, only keep text related to sentiment. For example, "There's a bloody issue with my damn account" should become "bloody damn."
Inclusion + prefix AP	Identify the sentiment of this customer support inquiry.	From this customer support inquiry, only keep text related to sentiment. For example, "There's a bloody issue with my damn account" should become "bloody damn."
Exclusion DP	-	From this customer support inquiry, remove any text related to the inquiry topic. For example, 'There's a bloody issue with my damn account' should become 'bloody damn.'
Classify DP	-	Classify the customer support inquiry below as one of the following categories: financial, contact, account, or distribution. Only list the category name with no additional explanation.
Classify AP	-	Classify the customer support inquiry below as one of the following categories: polite or rude. Only list the category name with no additional explanation.

Table 15: Prompts for Customer Support Dataset

Method	Prefix	LLM Preprocessing Prompt
Baseline	-	-
Prefix DP	Identify the programming language of this Stack Overflow question.	-
Prefix DP (with classes)	Identify the programming language of the question (typescript, javascript, python, bash/shell).	-
Prefix AP	Identify the question type of this StackOverflow post.	
Prefix AP (with classes)	Identify the question type (debugging, implementation, conceptual).	-
Inclusion DP	-	Rephrase this StackOverflow post to only keep text relevant to the programming language.
Inclusion + prefix DP	Identify the programming language of this Stack Overflow question.	Rephrase this StackOverflow post to only keep text relevant to the programming language.
Exclusion AP	-	From the following StackOverflow, remove any text that reveals a user's exact intent or needs. For example, "I need help debugging this stupid Python script. It's about using a hashmap for memoization." should become "A python script that uses a hashmap for memoization."
Inclusion AP	-	Rewrite the StackOverflow post to only include text essential to the general intent of the user's question.
Inclusion + prefix AP	Identify the question type of this StackOverflow post.	Rewrite the StackOverflow post to only include text essential to the general intent of the user's question.
Exclusion DP	-	Summarize the following StackOverflow post in a way that anonymizes the programming language or any technical tools. Keep the summary to one sentence and do not include details. For example, "I need help debugging this stupid Python script. It's about using a hashmap for memoization. What am I doing wrong?" should become "The user needs help debugging code."
Classify DP	-	"Classify the main programming language of the StackOverflow post below as one of the following: typescript, javascript, python, or bash/shell. Only list the category name with no additional explanation."
Classify AP	-	"Classify the main intent of the StackOverflow post below as one of the following: debugging, implementation, or conceptual. Only list the category name with no additional explanation."

Table 16: Prompts for StackOverflow Dataset

Method	Prefix	LLM Preprocessing Prompt
Baseline	-	-
Prefix DP	Identify the subreddit of the following Reddit comment.	-
Prefix DP (with classes)	Identify the subreddit (askreddit, leagueoflegends, buildapc, electronic_cigarette, or pcmasterrace) of the following Reddit comment.	-
Prefix AP	Identify the discourse act type of the given Reddit comment.	-
Prefix AP (with classes)	Identify the speech act (elaboration, agreement, answer, humor, question, announcement, disagreement, negative reaction, or appreciation) of the given Reddit comment.	-
Inclusion DP	-	Rewrite the following Reddit comment to only keep text strongly indicative of the subreddit. For example, "Let me clarify this point about ahri from league of legends" should become "ahri from league of legends."
Inclusion + prefix DP	Identify the subreddit of the following Reddit comment.	Rewrite the following Reddit comment to only keep text strongly indicative of the subreddit. For example, "Let me clarify this point about ahri from league of legends" should become "ahri from league of legends."
Exclusion AP	-	Rewrite the following Reddit comment in a way that does not reveal the type of discourse. For example, "Let me clarify this point about ahri from league of legends" should become "ahri from league of legends."
Inclusion AP		Summarize the following Reddit comment but only describe content relevant to the type of discourse. Keep the summary to one sentence. For example, 'Let me clarify this point about ahri from league of legends' should become 'The user is making a clarification'.

Table 17: Prompts for Reddit Posts Dataset

Method	Prefix	LLM Preprocessing Prompt
Inclusion + prefix AP	Identify the discourse act type of the Reddit comment summarized below.	Summarize the following Reddit comment but only describe content relevant to the type of discourse. Keep the summary to one sentence. For example, 'Let me clarify this point about ahri from league of legends' should become 'The user is making a clarification'.
Exclusion DP	-	Summarize the following Reddit comment in a way that completely anonymizes the subreddit. Keep the summary to one sentence. For example, 'Let me clarify this point about ahri from league of legends. Your gaming style will definitely mesh better with this character.' should become 'The user is making a clarification'.
Classify DP	-	Classify the subreddit of the Reddit comment below as part of one of the following subreddits: askreddit, leagueoflegends, buildapc, electronic_cigarette, or pcmasterrace. Only list the category name with no additional explanation.
Classify AP	-	Classify the Reddit comment below as one of the following categories: elaboration, agreement, answer, humor, question, announcement, disagreement, negative reaction, or appreciation. If none of these categories apply, classify as other. Only list the category name with no additional explanation.

Table 18: Prompts for Reddit Posts Dataset (cont.)