TnT-LLM: Text Mining at Scale with Large Language Models

Mengting Wan
Microsoft Corporation
Redmond, WA, United States
mengting.wan@microsoft.com

Tara Safavi
Microsoft Corporation
Redmond, WA, United States
tarasafavi@microsoft.com

Sujay Kumar Jauhar
Microsoft Corporation
Redmond, WA, United States
sjauhar@microsoft.com

Yujin Kim
Microsoft Corporation
Redmond, WA, United States
yujin.kim@microsoft.com

Scott Counts
Microsoft Corporation
Redmond, WA, United States
counts@microsoft.com

Jennifer Neville
Microsoft Corporation
Redmond, WA, United States
jenneville@microsoft.com

Siddharth Suri
Microsoft Corporation
Redmond, WA, United States
suri@microsoft.com

Chirag Shah*
University of Washington
Seattle, WA, United States
chirags@uw.edu

Ryen W. White
Microsoft Corporation
Redmond, WA, United States
ryenw@microsoft.com

Longqi Yang
Microsoft Corporation
Redmond, WA, United States
longqi.yang@microsoft.com

Reid Andersen
Microsoft Corporation
Redmond, WA, United States
reidandersen@microsoft.com

Georg Buscher
Microsoft Corporation
Redmond, WA, United States
georgbuscher@microsoft.com

Dhruv Joshi
Microsoft Corporation
Redmond, WA, United States
dhruv.joshi@microsoft.com

Nagu Rangan
Microsoft Corporation
Redmond, WA, United States
nrangan@microsoft.com

ABSTRACT

Transforming unstructured text into structured and meaningful forms, organized by useful category labels, is a fundamental step in text mining for downstream analysis and application. However, most existing methods for producing label taxonomies and building text-based label classifiers still rely heavily on domain expertise and manual curation, making the process expensive and time-consuming. This is particularly challenging when the label space is under-specified and large-scale data annotations are unavailable. In this paper, we address these challenges with Large Language Models (LLMs), whose prompt-based interface facilitates the induction and use of large-scale pseudo labels. We propose **TnT-LLM**, a two-phase framework that employs LLMs to automate the process of end-to-end label generation and assignment with *minimal human effort* for any given use-case. In the first phase, we introduce a zero-shot, multi-stage reasoning approach which enables LLMs to produce and refine a label taxonomy iteratively. In the second phase, LLMs are used as data labelers that yield training samples so that lightweight supervised classifiers can be reliably built, deployed, and served at scale. We apply **TnT-LLM** to the analysis of user intent and conversational domain for Bing Copilot (formerly Bing Chat), an open-domain chat-based search engine. Extensive experiments using both human and automatic evaluation metrics demonstrate that **TnT-LLM** generates more accurate and relevant label taxonomies when compared against state-of-the-art baselines, and achieves a favorable balance between accuracy and efficiency for classification at scale.

CCS CONCEPTS

• **Applied computing** → **Document analysis**.

KEYWORDS

Text Clustering, Text Classification, Large Language Models

*Work done while working at Microsoft.

1 INTRODUCTION

Text mining is the process of extracting useful information and insights from a large collection of textual data [10, 26]. Two central and interrelated tasks in text mining are **taxonomy generation**,
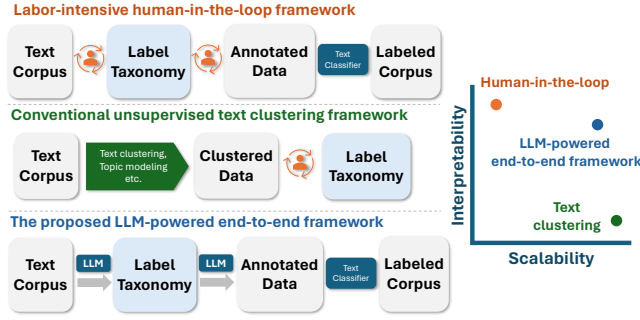
Figure 1: An illustration of the existing human-in-the-loop and unsupervised text clustering approaches as well as the proposed LLM-powered end-to-end label taxonomy generation and text classification framework (TnT-LLM).

which involves finding and organizing a set of structured, canonical labels that describe aspects of the corpus, and **text classification**, or the labeling of instances in the corpus using said taxonomy. Many use cases of interest to practitioners can be framed as the sequential application of these two tasks, especially when the label space is not well-defined or when exploring a new corpus: For example, sentiment analysis consists of devising a sentiment taxonomy (e.g., "happy", "sad") and classifying text content (e.g., social media posts, product reviews) with labels in this taxonomy. Likewise, intent detection consists of defining a set of intents (e.g., "book a flight", "buy a product") and classifying text content (e.g., chatbot transcripts, search queries) with the intent labels.

An established approach to these two tasks is to first employ domain experts to curate a label taxonomy [4, 19, 30], then gather human annotations on a small set of corpus sample using the hand-crafted taxonomy in order to train a machine learning model for text classification. While such human-in-the-loop approaches offer high *interpretability*, they face significant *scalability* challenges: They demand domain expertise and careful consideration of the granularity, coverage, and consistency of the labels [4], and manual annotation is time-consuming and costly, not to mention prone to errors and biases [27]. Moreover, the process must be repeated for each downstream use-case (e.g., sentiment analysis, intent detection, etc). Another line of work aims to solve these issues of scale via machine learning techniques like text clustering, topic modeling, and keyphrase mining. Such approaches flip the ordering of taxonomy generation and classification by first organizing the corpus sample into clusters in an unsupervised or semi-supervised fashion, then deriving the label taxonomy thereafter by describing the learned clusters. Such approaches scale better with the corpus size and use-cases, but describing text clusters in an interpretable and consistent way has proved challenging, so much so that is has been likened to "reading tea leaves" [5].

To address these challenges, in this paper we propose **TnT-LLM**, a novel framework that combines the interpretability of manual approaches with the scale of automatic text clustering and topic modeling. **TnT-LLM** is an end-to-end two-phase framework for joint **T**axonomy Generation **and T**ext Classification that relies on the unique strengths of instruction-following Large Language Models (**LLMs**) in both phases. First, in the taxonomy generation

phase, we devise a zero-shot multi-stage reasoning approach that prompts an LLM to produce and refine a label taxonomy iteratively with respect to the corpus for a given use-case (e.g., intent detection). Second, in the text classification phase, we adopt LLMs as data augmentors to scale up the creation of training data, which in turn is used to train lightweight classifiers capable of large-scale labeling. This framework is adaptable and modular, and can be customized to different use cases, text corpora, LLMs, and classifiers, while requiring little human intervention or input. In summary, our main contributions are as follows:

- We introduce **TnT-LLM**, an end-to-end two-phase framework to automate and scale the process of taxonomy generation and text classification with representative and interpretable labels.
- We present a series of quantifiable and traceable evaluation strategies to validate each stage of this framework, including *deterministic automatic* metrics, *human evaluation* metrics, as well as *LLM-based* evaluations.
- We use **TnT-LLM** to analyze conversations from Bing Copilot (formerly Bing Chat), a web-scale, multilingual, and open-domain conversational agent. Our results show that the proposed framework can produce more accurate and relevant label taxonomies compared to the state-of-the-art text clustering approaches. We also demonstrate that the lightweight label classifiers trained on LLM annotations can achieve comparable (and sometimes better) performance than directly using LLMs as classifiers, but with much higher scalability and model transparency. Through quantitative and qualitative analysis, we provide insights and recommendations for applying LLMs on large-scale text mining.

2 RELATED WORK

Taxonomy Generation. Prior work in taxonomy generation falls into manual and automatic approaches. Handcrafted taxonomies, beyond being expensive to construct, tend to be developed for specific downstream tasks (e.g., web search intent analysis [4, 19], chatbot intent detection [30]), or tied to the development of specific datasets [21, 24]. On the other hand, automated approaches scale better but either rely on term extraction from corpora to obtain labels, which may hinder interpretability and/or coverage [23, 32], or else require a set of seeds for the taxonomy in order to generate new labels [31]. **TnT-LLM**, in contrast, is automatic, abstractive (i.e., labels describe the corpus but need not be directly extracted from it), and does not require any seed labels. Moreover, **TnT-LLM** treats taxonomy generation and text classification as interrelated problems in an end-to-end pipeline, whereas prior work has tended to focus mainly on the quality of the taxonomy produced, without considering its downstream utility for classification.

Text Clustering and Topic Modeling. Text clustering and topic modeling "invert" the traditional approach of defining a label set, then applying the labels on the corpus. Given a set of documents, such approaches first group the documents into topical clusters using various definitions of textual similarity, then post-hoc label or summarize the clusters [1, 28]. While traditional approaches in theory accomplish the same goals as **TnT-LLM**, they suffer due to a lack of interpretability [5], as they typically do not assign intelligible labels to clusters. More recently, attempts have been made to overcome these problems by using LLMs for topic modeling [17, 29],
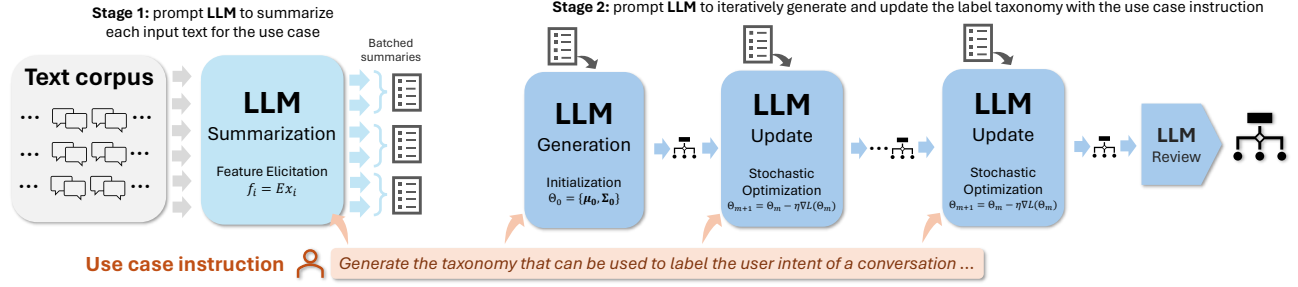
Figure 2: An illustration of the LLM-powered taxonomy generation phase (Phase 1).

though these approaches still require supervision through either a predefined taxonomy [29] or a seed set of topics [17].

LLMs as Annotators. Recent work has explored using LLMs to replace human annotation for labor-intensive tasks such as search relevance quality labeling [27], topic and stance detection [8], and various computational social science labeling tasks [33]. These studies have found that, in general, LLMs perform on par or even better than crowd-workers [14], often at a fraction of the cost. In the same vein, we explore using LLMs as annotators for text classification, although our main goal is to scale the process by distilling LLMs' label-specific capabilities into more efficient, lightweight classifiers.

3 METHOD

We begin with a high-level overview of **TnT-LLM**, our proposed two-phase framework for 1) **LLM-powered taxonomy generation** and 2) **LLM-augmented text classification**. In the first phase, we sample a small-scale representative subset of a corpus and perform zero-shot multi-stage taxonomy generation in an iterative manner inspired by stochastic gradient descent [3]. In the second phase, we sample a larger dataset and leverage LLMs with the taxonomy produced by Phase 1 to classify each instance. These LLM labels are then treated as "pseudo-labels" for training a lightweight text classifier. Once training is complete, the lightweight classifier is deployed to label the entire corpus offline, and may also serve for online real-time classification.

3.1 Phase 1: Taxonomy Generation

Phase 1 of **TnT-LLM** is inspired by the classic mixture model clustering process [16], but implemented in a prompt-based manner. We leverage a "stochastic optimization" approach [18] to iteratively update the intermediate taxonomy outcome, so that a large and dynamic corpus sample can be effectively handled. Depending on the desired granularity of the taxonomy, we suggest using a "small-to-medium" corpus sample that is representative of the corpus in this phase, such that the sample size is sufficient to capture the diversity of the corpus, but not too large to incur unnecessary costs.

- **Stage 1: Summarization.** In order to normalize all text samples and extract their most salient information, we first generate concise and informative summaries of each document in the sample. Specifically, we prompt an LLM to summarize each document by providing a short blurb about the intended use-case for the summary (e.g., intent detection) and a target summary length (e.g., 20

words); the full prompt template is provided in Prompt 1 in the supplemental details. This stage helps reduce the size and variability of the input documents while also extracting the aspects of the document most relevant to the use-case, which we find is especially important for label spaces that are not evident from surface-level semantics (e.g., user intent). Note that this stage is relatively fast, as it may be executed concurrently for each input document with a cost-efficient LLM like GPT-3.5-Turbo.

- **Stage 2: Taxonomy Creation, Update, and Review.** We next create and refine a label taxonomy using the summaries from the previous stage. Similar to SGD, we divide the summaries into equal-sized minibatches. We then process these minibatches with three types of zero-shot LLM reasoning prompts in sequence. The first, an *initial generation prompt*, takes the first minibatch and produces an initial label taxonomy as output. The second, a *taxonomy update prompt*, iteratively updates the intermediate label taxonomy with new minibatches, performing three main tasks in each step: 1) evaluating the given taxonomy on the new data; 2) identifying issues and suggestions based on the evaluation; and 3) modifying the taxonomy accordingly. Finally, after the taxonomy has been updated a specified number of times, we apply a *review prompt* that checks the formatting and quality of the output taxonomy, of which the output is regarded as the final taxonomy output by Stage 1. In all three prompts, we provide the use-case instruction, which specifies the goal and the format of the desired label taxonomy (e.g., the desired number of labels, the target number of words per label), alongside the minibatch. The full prompt templates are provided in Prompt 2–4 in the supplemental details.

Notice that this process naturally lends itself to hierarchy: After a first round of taxonomy generation, we can rerun Stage 2 for each subgroup of categorized samples to create new, more granular levels in the taxonomy. An overview of our proposed approach is presented in Figure 2.

Connection to Mixture Models & Stochastic Optimization. Here we present an analogy between our pipeline and the Mixture Model family (e.g., Gaussian Mixture Model) for text clustering. We assume each text data point (x_i) follows a mixture distribution $x_i \sim \sum w_k \mathcal{N}(\mu_k, \Sigma_k)$, where $\mathcal{N}(\mu_k, \Sigma_k)$ defines the distribution of the k-th component (a Gaussian distribution with a mean μ_k and variance Σ_k). Given a corpus sample $\{x_i\}$, this mixture can be learned through Maximum Likelihood Estimation (MLE), equivalent
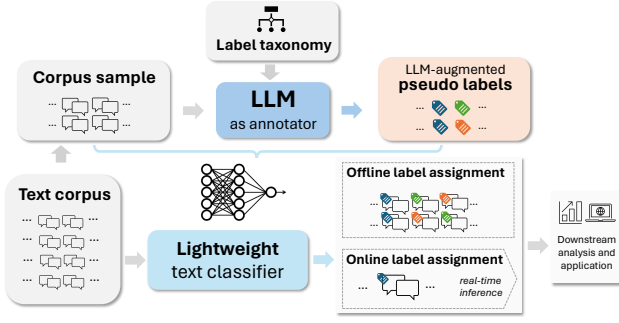
Figure 3: An illustration of the LLM-augmented text classification phase (Phase 2).

to minimizing the negative of the log-likelihood loss, i.e.,

$$\max \prod_i \Big(\sum w_k \mathcal{N}(\mu_k, \Sigma_k; x_i) \Big)$$
$$\Leftrightarrow \min \; - \sum_i \log \Big(\sum w_k \mathcal{N}(\mu_k, \Sigma_k; x_i) \Big) \Leftrightarrow \min \sum_i \mathcal{L}(\Theta, x_i). \quad (1)$$

Mapping back to our prompt-based approach, we take a corpus sample and a use-case instruction as input. Our goal is to "learn" a taxonomy that is relevant to the instruction and best fits the input corpus sample; this taxonomy must consist of category labels with names and brief descriptions. We can represent our desired label taxonomy as a parameter set $\Theta = \{\mu, \Sigma\}$, following the definition of the mixture model, where $\mu = \{\mu_k\}$ are the names of labels k which represent the "cluster centroids," and $\Sigma = \{\Sigma_k\}$ are the descriptions that specify the "shape" of cluster k. We assume the mixture weights (w_k) are implicitly captured by the LLM that generates the label taxonomy in this study. We can then map our taxonomy creation and refinement stages to stochasic optimization as follows:

- **Stage 1: Feature Representation.** Our summarization stage is analogous to the featurization step in classic machine learning, where raw text inputs are projected onto a vector space via a feature transformation such as an embedding model. In our case, the output summary of each data point can be viewed as a concise and informative feature representation of the original text (x_i).

- **Stage 2: Stochastic Gradient Descent.** The main taxonomy creation and update stage resembles prompt optimization with Stochastic Gradient Descent (SGD) [18], where the generation prompt is used to initialize the taxonomy (i.e., the parameters Θ_0), which is then optimized via SGD through the update prompt-chain. In each update prompt, we assess how the current taxonomy (Θ_m) fits the given batch of data (i.e., calculating the loss function defined in Eq. (1)), then analyze and "backpropagate" the errors to update the taxonomy, i.e., $\Theta_{m+1} = \Theta_m - \eta \nabla \mathcal{L}(\Theta_m)$, where η refers to the learning rate which we assume is implicitly adjusted by the LLM.

3.2 Phase 2: LLM-Augmented Text Classification

After the taxonomy is finalized, we next train a text classifier that can be reliably deployed to perform label assignments at very large-scale and in real-time. Following recent work that shows the strengths of LLMs as annotators of training data [8, 14], we propose to leverage LLMs to obtain a "pseudo-labeled" corpus set

using the taxonomy yielded in Phase 1, then use these labels to train more efficient classifiers at scale. Specifically, we prompt an LLM to infer the primary label (as a multiclass classification task) and all applicable labels (as a multilabel classification task) on a "medium-to-large" scale corpus sample that covers the range of labels in the taxonomy, creating a representative training dataset that can be used to build a lightweight classifier, such as a Logistic Regression model or a Multilayer Perceptron classifier. In this way, we can induce "pseudo labels" from the LLM classifier and transfer its knowledge to a more efficient and manageable model that can be deployed and served at scale. An illustrative figure of this phase is presented in Figure 3.

4 EVALUATION SUITE

Due to the unsupervised nature of the problem we study and the lack of a benchmark standard, performing quantitative evaluation on end-to-end taxonomy generation and text classification can be challenging. We therefore design a suite of strategies to evaluate **TnT-LLM**. Our evaluation strategies may be categorized into three buckets, depending on the type and source of the evaluation criteria. The three categories are as follows:

- **Deterministic automatic evaluation**: This type of approach is scalable and consistent, but requires well-defined, gold standard rules and annotations. It is less applicable for evaluating the abstract aspects studied in this paper, such as the quality and usefulness of a label taxonomy.

- **Human evaluation**: These approaches are useful for evaluating the abstract aspects that the automatic evaluations cannot address. However, they are also time-consuming, expensive, and may encounter data privacy and compliance constraints.

- **LLM-based evaluations**: Here, LLMs are used to perform the same or similar tasks as human evaluators. This type of evaluation is more scalable and cost-effective than human evaluation, albeit potentially subject to biases and errors if not applied properly. We therefore aim to combine and validate LLM-based evaluation with human evaluation metrics on small corpora so that we can extrapolate conclusions with sufficient statistical power.

4.1 Phase 1 Evaluation Strategies

Following prior studies [22, 29], we evaluate a label taxonomy on three criteria: Coverage, accuracy, and relevance to the use-case instruction. Note that we require implementing the *native* primary label assignment to apply these metrics. For clustering-based methods, this is instantiated through the clustering algorithm. For **TnT-LLM**, this is done by a label assignment prompt as described in Section 3.2. We also note that the label accuracy and use-case relevance metrics discussed here are applicable to both **human** and **LLM** raters.

Taxonomy Coverage. This metric measures the comprehensiveness of the generated label taxonomy for the corpus. Conventional text clustering approaches (e.g., embedding-based k-means) often achieve 100% coverage by design. In our LLM-based taxonomy generation pipeline, we add an 'Other' or 'Undefined' category in the label assignment prompt by design and measure the proportion of data points assigned to this category. The lower this proportion, the higher the taxonomy coverage.

Label Accuracy. This metric quantifies how well the assigned label reflects the text data point, relative to other labels in the same taxonomy. Analogous to mixture model clustering, the primary label should be the most probable one given the text. We assume human and LLM raters can assess the label fit by its name and description. We treat accuracy as a pairwise comparison task: for each text, we obtain the primary label and a random negative label from the same taxonomy, and ask a rater to choose the more accurate label based on their names and descriptions.[1] If the rater correctly identifies the positive label, we consider it as a "Hit" and report the average hit rate as the label accuracy metric. We do not explicitly evaluate the overlap across category labels and rather expect it to be implicitly reflected in the pairwise label accuracy metric.

Relevance to Use-case Instruction. This metric measures how relevant the generated label taxonomy is to the use-case instruction. For example, "Content Creation" is relevant to an instruction to "understand user intent in a conversation", while "History and Culture" is not. We operationalize this as a binary rating task: for each instance, we provide its primary label name and description to a human or LLM rater, and ask them to decide if the label is relevant to the given use-case instruction or not. Note that we instruct the rater to use the presented instance as the context, and rate the relevance conditioned on the label's ability to accurately describe some aspect of the text input. The goal of this metric is not to evaluate the label accuracy, but rather to rule out the randomness introduced by taxonomies that are seemingly relevant to the use-case instruction, but irrelevant to the corpus sample – and therefore useless for downstream applications.

4.2 Phase 2 Evaluation Strategies

To quantitatively evaluate text classification, we create a benchmark dataset with reliable ground-truth annotations as follows:

Task and Annotation Reliability. We first assess the reliability of the label assignment task and the human annotations by involving multiple human annotators and calculating the inter-rater agreement (Cohen's Kappa [6] between two raters and Fleiss' Kappa [7] among multiple raters). We then resolve disagreements between human annotations by either voting or deliberation, and obtain a consensus human annotation for each instance. Then we use an LLM as an additional annotator to perform the same label assignment task, and measure the agreement between the LLM annotation and the consensus human label. Intuitively, this agreement captures how well the LLM is aligned with (the majority of) human annotators and how reliable it is for this label assignment task.

Classification Metrics. We apply both human and LLM annotations on a small-scale corpus sample and calculate the conventional multiclass and multilabel classification metrics (e.g., Accuracy, F1) with human annotations as the ground truth. These metrics evaluate how the label classifier is aligned with human preferences on a small subset of the corpus. We then apply the LLM annotator on a larger-scale corpus sample and leverage the resulting annotations as the oracle to calculate the same classification metrics. These metrics enable a comprehensive diagnosis of the label classifier

performance at scale on different aspects of the corpus, such as domains, languages, and time ranges.

In practice, we recommend leveraging both human evaluation and LLM-based metrics as a holistic evaluation suite, while also taking into account the task and annotation reliability. This approach can help us identify and mitigate the possible bias that may arise from either method or be affected by the task complexity, and enable us to scale up the evaluation and annotation to a large corpus sample with confidence, thus obtaining more robust and informative evaluation results.

5 EXPERIMENTS

We showcase the utility of **TnT-LLM** for two text mining tasks of special interest in today's LLM era: **User intent detection** and **conversational domain labeling** over human-AI chat transcripts.

5.1 Data

Our conversation transcripts are taken from Microsoft's Bing Consumer Copilot system, which is a multilingual, open-domain generative search engine that assists users through a chat experience. We randomly sample 10 weeks of conversations from 8/6/2023 to 10/14/2023, with 1k conversations per week for Phase 1, where we perform a random 60%-20%-20% split for "learning" the label taxonomy, validation, and testing respectively. We then sample another 5k conversations per week from the same time range for Phase 2, and apply the same train/validation/test data split.

We perform two steps of filtering to ensure the quality and privacy of the data. First, we apply an in-house privacy filter that scrubs all personal information (e.g., addresses, phone numbers) from the original conversation content. Second, we apply a content filter that removes all conversations that contain harmful or inappropriate content that should not be exposed to annotators or downstream analyses. After applying these filters, we obtain 9,592 conversations for Phase 1 and 48,160 conversations for Phase 2. We leverage the FastText language detector [11, 12] to identify the primary language of each conversation, where we find around half of the conversations in our corpus are in English.

In the remainder of this section, we will report results on the following datasets:

- **BingChat-Phase1-L-Multi**: The test set used in the taxonomy generation phase, which includes around 2k conversations.
- **BingChat-Phase2-L-Multi**: The test set used in the label assignment phase, which includes around 10k conversations.

Besides the above datasets, we also reserve two separate English-only conversation datasets to perform human evaluations, with the same privacy and content filter applied.

- **BingChat-Phase1-S-Eng** includes 200 English conversations to evaluate label taxonomy.
- **BingChat-Phase2-S-Eng** includes 400 English conversations to evaluate label assignment.

5.2 Taxonomy Generation

5.2.1 Methods. To evaluate the effectiveness of **TnT-LLM**, we compare it with baseline methods that rely on embedding-based clustering to group conversations and then assigns LLM-generated

---

[1]Raters are also offered a "None" option besides the pair, but are instructed to minimize the use of it.

labels to each cluster. We use two state-of-the-art LLMs, **GPT-4 (0613)** and **GPT-3.5-Turbo (0613)**, as label generators and evaluators, and two different embedding methods, **ADA2**[2] and **INSTRUCTOR-XL** [25], to represent the conversations. The methods considered in our experiments are as follows:

- **GPT-4 (TnT-LLM)**: the proposed **TnT-LLM** with GPT-4 to perform label taxonomy generation and assignment.
- **GPT-3.5 (TnT-LLM)**: the proposed **TnT-LLM** with GPT-3.5-Turbo to perform label taxonomy generation and assignment.
- **ADA2 + GPT-4**: the embedding-based clustering approach where conversations are represented via **ADA2** and K-means algorithm is applied to generate clusters. We randomly sample 200 conversations within each cluster, prompt GPT-4 to summarize each conversation, then ask it to produce a label name and description from these summaries, conditioned on the use-case instruction.
- **ADA2 + GPT-3.5-Turbo**: similar to the above method, with GPT-3.5-Turbo as the label generator.
- **INSTRUCTOR-XL + GPT-4**: similar to the above embedding-based methods, with **INSTRUCTOR-XL** and GPT-4 as the underlying embedding and the label generator respectively.
- **INSTRUCTOR-XL + GPT-3.5-Turbo**: similar to the above method, with GPT-3.5-Turbo as the label generator.

Note that all the taxonomies evaluated in this section are fully automatic and do not involve any human intervention.

5.2.2 Implementation Details. We instruct our LLMs to generate 10 intent categories and 25 domain categories for taxonomy generation. Likewise, we learn 10 intent clusters and 25 domain clusters with our embedding-based baselines. We use a minibatch size of 200 for our proposed taxonomy generation pipeline. We also apply a minibatch version of the K-means algorithm in all embedding-based clustering approaches, where the same batch size is used with a K-means++ [2] initialization. We run 10 different trials of the clustering algorithm and select the best one based on the Silhouette coefficient [20] on the validation set. We also devise a "model" selection prompt, which takes a batch of conversation summaries, multiple label taxonomies, a use-case instruction as input, then outputs the index of the taxonomy that best fits the data and the instructional desiderata. We then run **TnT-LLM** 10 trials and select the best outcome based on its performance on the validation set.

Human Evaluation. To evaluate the quality of generated taxonomies from methods listed above, three of the authors performed the label accuracy and use-case relevance tasks; each conversation was evaluated by all three raters. While raters possessed a high degree of familiarity with the Bing Copilot system, as well as the desired use-cases, they were unaware of the correspondence between methods and their generated labels. The position of the options in the pairwise comparison label accuracy task is also fully randomized. We also use two LLM systems, GPT-4 and GPT-3.5-Turbo, to perform the same evaluation tasks as the human raters. However, we notice that the LLM systems tend to exhibit a position bias [15] for the pairwise comparison task, where they favor one option over another based on its position in the prompt. This bias is more evident when the taxonomy quality is low and the task is more

Metric	Use Case	Among Humans		LLM vs. Human	
		Overall (Fleiss)	Avg. pairwise (Cohen)	GPT-3.5-Turbo (Cohen)	GPT-4 (Cohen)
Accuracy	Intent	0.476*	0.477*	0.376	0.558*
	Domain	0.478*	0.484*	0.260	0.578*
Relevance	Intent	0.466*	0.481*	0.333	0.520*
	Domain	0.379	0.399	0.177	0.288

Table 1: Inter-rater reliability (Fleiss' Kappa and Cohen's Kappa) among human raters and between LLM raters and the resolved human rating through majority voting. Agreement considered as *moderate* and above (> 0.4) are highlighted with *. Evaluation is performed on BingChat-Phase1-S-Eng.

challenging. To mitigate this, we average the results over multiple runs with randomized positions of the options in our experiments.

5.2.3 Results. We first calculate the **coverage** of the LLM-generated taxonomies on the **BingChat-Phase1-L-Multi** dataset, where both LLM systems achieve very high coverage (>99.5%) on both user intent and conversational domain taxonomies.

We then conduct the accuracy and relevance evaluation tasks to assess the quality of the taxonomies generated by different methods on the small English-only evaluation dataset **BingChat-Phase1-S-Eng**. We report the inter-rater agreement (Cohen's Kappa [6] between two raters and Fleiss' Kappa [7] among multiple raters) in Table 1. The agreement is *moderate* ($\kappa > 0.4$) on intent and domain accuracy as well as intent relevance, while the agreement on domain relevance is *fair* (*Fleiss'* $\kappa = 0.379$).[3] Interestingly, for the tasks with *moderate* agreement, the GPT-4 evaluator agrees more with the human majority than the humans do among themselves. This suggests that GPT-4 can be a consistent and reliable evaluator.

Figure 4a shows the main results on **label accuracy** and **use case relevance** from human evaluations on **BingChat-Phase1-S-Eng**. We observe our **TnT-LLM** using GPT-4 outperforms other methods in most cases. Compared to GPT-4, we find that GPT-3.5-Turbo tends capture conversation topics (domains) well, but often fails to generate labels that are aligned with the user intent instruction. Likewise, we notice that some embedding methods (**ADA2 + GPT-4**, **INSTRUCTOR-XL + GPT-4**) perform well in terms of producing accurate domain labels, on par with **TnT-LLM** instantiated with GPT-3.5-Turbo, but fail to capture the user intent behind the conversations. This is likely because the domain labels reflect the topical theme of the conversations, which can be easily derived from the semantic information captured by unsupervised embeddings, while intent labels require deeper reasoning and understanding of the use-case instruction.

With regard to our baselines, we find that GPT-4 consistently outperforms GPT-3.5-Turbo in producing more accurate labels when using the same embedding method for clustering. For the intent use-case, GPT-4 generates more relevant labels than GPT-3.5-Turbo, while the difference is less noticeable for the domain use case; again, this may be because GPT-3.5-Turbo is better at capturing topical information in conversations than reasoning about user intent.

---

[2]https://openai.com/blog/new-and-improved-embedding-model

[3]Note that these evaluation tasks are cognitively challenging, especially for low-quality taxonomies (e.g., from some baseline methods).
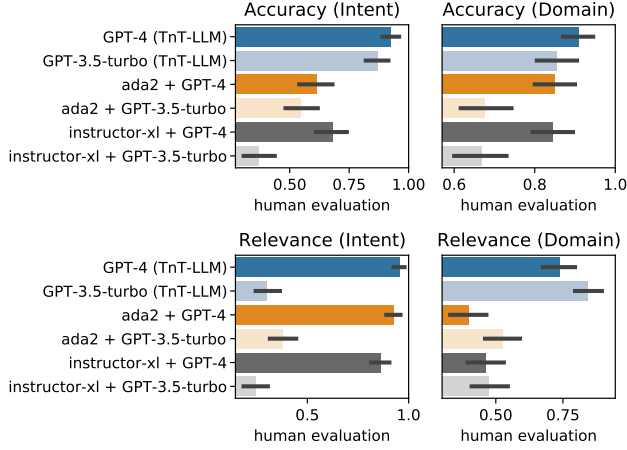
(a) Human evaluation results on BingChat-Phase1-S-Eng.
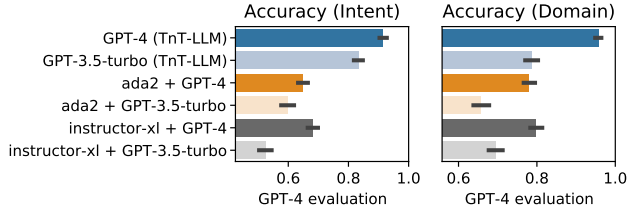
(b) GPT-4 evaluation results on BingChat-Phase1-L-Multi.

Figure 4: Taxonomy evaluation results on BingChat-Phase1-S-Eng from human raters and the GPT-4 rater, where error bars indicate 95% confidence intervals.

Finally, given the high agreement between GPT-4 and human raters on the label accuracy task, we use GPT-4 to evaluate the label accuracy on the larger multilingual dataset **BingChat-Phase1-L-Multi** (Figure 4b). We observe similar patterns as those in our human evaluation, where our **TnT-LLM** achieves the highest accuracy, and in particular the instantation that uses GPT-4.

5.3 LLM-Augmented Text Classification

At the end of the label taxonomy generation phase, we conduct a lightweight human calibration [22] on the intent taxonomy and domain taxonomy generated from **TnT-LLM** with GPT-4 to improve their clarity. These calibrated taxonomies are then utilized in the label assignment phase. The full label and description texts of each taxonomy are provided in Table 5a and Table 5b. As a reminder, our main goal in this section is to compare how distilled lightweight classifiers trained on LLM labels compare to a full LLM classifier; our goal is to achieve a favorable tradeoff of accuracy and efficiency compared to a more expensive but potentially more powerful LLM.

5.3.1 Methods. We apply GPT-4 as an automated annotator to assign both the primary label and any other relevant labels to each conversation in the corpus. We then train classifiers based on the GPT-4 annotated training and validation sets. We extract features from each conversation using two embedding methods: **ada2** and **Instructor-XL**. For each embedding method, we train three types of classifiers with the GPT-4 labels: **Logistic Regression**, the

Metric	Use Case	Among Humans		LLM vs. Human
		Overall (Fleiss)	Avg. pairwise (Cohen)	GPT-4 (Cohen)
Primary Label	Intent	0.553*	0.559*	0.572*
	Domain	0.624**	0.624**	0.695**
All Labels (exact match)	Intent	0.422*	0.427*	0.271
	Domain	0.467*	0.467*	0.102

Table 2: Inter-rater reliability (Fleiss' Kappa and Cohen's Kappa) among human annotators and between LLM annotations and the resolved human annotations. Agreement considered as *moderate* ($(0.4, 0.6]$) are highlighted with *, *substantial* and above (> 0.6) are highlighted with **.

gradient boosting **LightGBM** [13], and a two-layer **MultiLayer Perceptron (MLP)** [9]. We use multinomial logit in **logistic regression** for the primary label classification, and a standard 'one-vs-all' scheme for the multilabel classification with all three classifiers.

Additionally, four of the authors manually labeled 400 English conversations (**BingChat-Phase2-S-Eng**) with the given intent and domain taxonomy. Each conversation was labeled by three annotators, and the majority vote determined the final labels. For a few conversations (<10%), where all three annotators disagreed on the primary label the fourth annotator was used as a tie-breaker.

We thus obtain two annotated test sets: **BingChat-Phase2-S-Eng** with 400 English conversations with both human and GPT-4 annotations, and **BingChat-Phase2-L-Multi** with around 10k conversations with GPT-4 annotations only.

5.3.2 Results. We first evaluate the agreement between annotators to assess the task complexity and reliability. As Table 2 shows, human annotators have *substantial* agreement on the primary domain label ($\kappa > 0.6$), and *moderate* agreement on the primary intent label (*Fleiss' κ* = 0.553). Both of these values indicate a high degree of mutual understanding among raters and clarity in the instructions and taxonomies. We also note that the domain taxonomy has more categories (25) than the intent taxonomy (10). One might expect a larger taxonomy to be more difficult to comprehend, but we find the smaller intent taxonomy to be more challenging for humans to agree on. We attribute this to the task complexity and ambiguity, as it requires more reasoning; this observation aligns well with our observation in the previous evaluation that GPT-4 greatly outperforms GPT-3.5-Turbo on intent detection, as GPT-4 is generally considered to be a stronger reasoner.

Similar to the label accuracy evaluation (Table 1), GPT-4 agrees more with the resolved human labels than humans do among themselves on the primary label assignment. We observe that human agreement on all applicable labels is *moderate* ($\kappa > 0.4$) with both intent and domain taxonomies, which is surprisingly good considering such an agreement is calculated based on exact match (i.e., an agreement is counted only if all selected labels are matched). However, the agreement between GPT-4 and human annotations on this task is much lower. A closer inspection reveals that GPT-4 tends to be more liberal than humans on label assignment, applying all relevant categories, resulting in a low precision but high recall.

Oracle	Human Annot.		GPT-4 Annot.		
	Accur.	F1 macro	Accuracy		
			All	English	Non-Eng.
User Intent					
GPT-4	0.655	**0.640**			
ADA2 +					
LogisticReg	**0.658** □	0.639	**0.746**	**0.763** +2.3%	**0.725** -2.7%
LightGBM	0.642 □	0.536	0.702	0.716 +2.0%	0.686 -2.3%
MLP	0.658 □	0.602	0.744	0.762 +2.4%	0.722 -2.9%
INSTRUCTOR-XL +					
LogisticReg	0.655 □	0.611	0.687	0.745 +8.4%	0.619 -9.9%
LightGBM	0.602 ↓	0.455	0.652	0.705 +8.1%	0.589 -9.6%
MLP	0.650 □	0.593	0.691	0.750 +8.0%	0.621 -10.1%
Conversation Domain					
GPT-4	0.638	**0.603**			
ADA2 +					
LogisticReg	0.640 □	0.573	**0.705**	**0.733** +3.9%	**0.673** -4.6%
LightGBM	0.560 ↓	0.476	0.633	0.656 +3.8%	0.605 -4.4%
MLP	**0.650** □	0.583	0.703	0.731 +4.1%	0.669 -4.8%
INSTRUCTOR-XL +					
LogisticReg	0.622 □	0.562	0.639	0.711 +11.3%	0.553 -13.3%
LightGBM	0.588 ↓	0.505	0.583	0.646 +10.9%	0.508 -12.8%
MLP	0.648 □	0.569	0.639	0.712 +11.4%	0.553 -13.4%

Table 3: Lightweight distilled classifiers achieve competitive performance compared to a full GPT-4 classifier on Phase 2: Primary label classification results on 1) BingChat-Phase2-S-Eng with human annotations as the oracle and 2) BingChat-Phase2-L-Multi with GPT-4 annotations as the oracle. For BingChat-Phase2-S-Eng, we mark whether the classifier results are significantly higher (↑), lower (↓), or insignificant (□) than GPT-4 by paired t-test ($p < 0.05$). For BingChat-Phase2-L-Multi, we indicate the percentage changes for English and non-English conversations compared to the overall result for each classifier.

We then evaluate the classification performance of the distilled embedding-based classifiers on two datasets: **BingChat-Phase2-S-Eng**, where human annotations are the oracle, and **BingChat-Phase2-L-Multi**, where GPT-4 annotations are the oracle. The results for the primary label classification are presented in Table 3, where we observe that lightweight embedding-based classifiers can achieve promising results. In particular, **ADA2** embeddings achieve strong results with logistic regression; nonlinearity does not seem to improve performance significantly in most cases. When using human annotations as the gold standard, we find that the performance of these lightweight models are comparable to, and sometimes slightly better than, directly using GPT-4 as a classifier on **BingChat-Phase2-S-Eng**. We also perform evaluation on the multilingual test set **BingChat-Phase2-L-Multi**, where GPT-4 annotations are considered as oracle. We observe the performance on non-English conversations is lower than that on English conversations (Table 3), especially on the **INSTRUCTOR** embedding, indicating the importance of choosing an appropriate embedding method that suits the characteristics of the corpus.

	Accur.	Micro			Macro		
		Precision	Recall	F1	Precision	Recall	F1
User Intent							
GPT-4	0.320	0.518	**0.743**	0.610	0.613	**0.644**	**0.537**
ADA2 +							
LogisticReg	0.388 ↑	0.574 ↑	0.736 □	**0.645**	0.593	0.607	**0.537**
LightGBM	0.380 ↑	0.587 ↑	0.669 ↓	0.626	0.610	0.486	0.456
MLP	**0.418** ↑	0.599 ↑	0.657 ↓	0.627	**0.626**	0.513	0.499
INSTRUCTOR-XL +							
LogisticReg	0.358 ↑	0.559 ↑	0.688 ↓	0.617	0.583	0.540	0.51
LightGBM	0.335 □	0.557 ↑	0.644 ↓	0.597	0.571	0.479	0.465
MLP	0.410 ↑	**0.606** ↑	0.642 ↓	0.623	0.623	0.480	0.495
Conversation Domain							
GPT-4	0.110	0.442	**0.753**	0.557	0.565	**0.687**	0.576
ADA2 +							
LogisticReg	0.188 ↑	0.493 ↑	0.732 ↓	**0.589**	0.644	0.624	**0.585**
LightGBM	0.182 ↑	0.469 ↑	0.576 ↓	0.517	0.621	0.440	0.452
MLP	0.242 ↑	0.532 ↑	0.625 ↓	0.575	0.667	0.490	0.509
INSTRUCTOR-XL +							
LogisticReg	0.210 ↑	0.495 ↑	0.714 ↓	0.585	**0.655**	0.602	0.574
LightGBM	0.172 ↑	0.479 ↑	0.592 ↓	0.530	0.586	0.453	0.469
MLP	**0.262** ↑	**0.550** ↑	0.602 ↓	0.575	0.738	0.475	0.511

Table 4: Lightweight distilled classifiers perform on par with or better than GPT-4 on multilabel classification: Results on BingChat-Phase2-S-Eng using human-annotated gold labels.

On the multilabel classification task (Table 4), we observe that the distilled classifiers achieve higher precision at the expense of some recall compared to GPT-4. Here, nonlinearity also seems to help more, as MLP-based classifiers achieve the highest accuracy and precision.

5.4 Cost Analysis

We base on the domain taxonomy experiments and provide a brief cost analysis with the following assumptions: 550 tokens per conversation, 20 tokens per conversation summary, 25*30 tokens per generated taxonomy, and 200 conversation summaries in each taxonomy generation batch. Using the publicly available pricing information from Azure Open AI (AOAI)[4], along with the token estimates derived from our experiments, we approximate the cost for each LLM or embedding call as the follows:

	Model	Input Tokens	Output Tokens	Cost (per call)
Phase 1: Summarization	GPT-4	780	80	$0.056
Phase 1: Taxon. Generation	GPT-4	5000	1750	$0.510
Phase 2: Label Assignment	GPT-4	1100	130	$0.082
	Ada2	550	-	$0.00006

We compare 1) **TnT-LLM** with a distilled **embedding-based classifier** and 2) directly serving an **LLM-based classifier** with a generated taxonomy and consider the following two types of costs:

---

[4]GPT-4 (32k, 0613): $0.06 per 1k input tokens, $0.12 per 1k output tokens; Ada2: $0.0001 per 1k input tokens.

- **One-off development cost** ($C_{dev.}$). For an **LLM-based classifier**, this cost only includes Phase 1 taxonomy generation ($C_{dev.} = C_{taxon.}$). For **TnT-LLM** with an **embedding-based classifier**, this includes both taxonomy generation and classifier development ($C_{dev.} = C_{taxon.} + C_{annot.}$). Typically a larger corpus sample ($N_{annot.}$) is required for Phase 2 compared to Phase 1 ($N_{taxon.}$) and the expensive taxonomy generation process in Phase 1 can be effectively mitigated via batching. Therefore, the primary cost for developing **TnT-LLM** comes from Phase 2 LLM annotations, where $C_{annot.} = N_{annot.}.(c_{llm}+c_{embed.})$, with c_{llm} and $c_{embed.}$ representing the costs per LLM classifier and embedding API calls, respectively.

- **Variable serving cost** ($C_{serv.}$) relates to the deployment for downstream tasks and scales with the size of the corpus ($N_{serv.}$) to be served, i.e., $C_{serv.} = N_{serv.}c_{llm}$ for an **LLM-based** classifier and $C_{serv.} = N_{serv.}c_{embed.}$ for an **embedding-based** classifier.

We now contrast their total costs ($C = C_{dev} + C_{serv}$) here:

$$\frac{C_{embed.}}{C_{llm}} == \frac{\frac{C_{taxon.}}{N_{serv.}c_{llm}} + \frac{N_{annot.}}{N_{serv.}}\frac{c_{embed.}}{c_{llm}} + \frac{c_{embed.}}{c_{llm}} + \frac{N_{annot.}}{N_{serv.}}}{\frac{C_{taxon.}}{N_{serv.}c_{llm}} + 1} \qquad (2)$$

This ratio is influenced by both the serving corpus size $N_{serv.}$ and the unit cost ratio $\frac{c_{embed.}}{c_{llm}}$ ($\ll 1$). When $N_{serv.} \gg N_{annot.}$, we can easily achieve $\frac{C_{embed.}}{C_{llm}} < 1$, suggesting that lightweight classifiers are more cost-effective when the serving corpus size is substantially large. As $N_{serv.}$ increases, this ratio then converges to the unit cost ratio $\frac{c_{embed.}}{c_{llm}}$, indicating the ongoing need for efficient embeddings.

To illustrate these observations, we fix the development corpus sizes at $N_{taxon.} = 10k$ and $N_{annot.} = 50k$, vary the serving corpus size $N_{serv.}$, and present the total costs for both embedding-based and LLM-based classifiers in Figure 5. The results align with our cost analysis, demonstrating the relative cost-effectiveness of each approach under different serving corpus sizes.
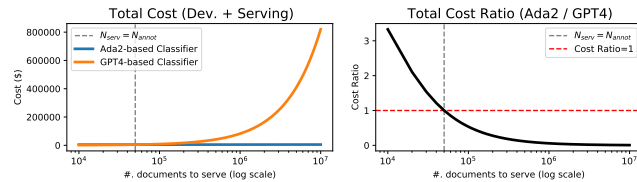

Figure 5: Total cost comparison between serving embedding-based (Ada2) and LLM-based (GPT-4) classifiers.

5.5 Summary of Findings and Suggestions

We have shown that our novel **TnT-LLM** framework is capable of generating high-quality label taxonomies from unstructured text corpora with very little human instruction or intervention. In our evaluation of this approach on real-world AI chat conversations, we demonstrated that it can be used to find structure and organization in unstructured text. Our method outperforms the conventional embedding-based clustering approach, especially when deeper reasoning beyond surface-level semantics is required. Finally we found that while embedding-based clustering can still be effective, it is more susceptible to modeling choices or artifacts, such as cluster granularity and alignment of use-case with inputs.

We further explored the use of LLMs as raters or evaluators, demonstrating that they effectively approximate the collective opinion of humans on some evaluation tasks. Additionally, we found that LLMs excel at single-choice questions (e.g., pairwise label accuracy evaluation task) where they are forced to indicate preference on one option over another, but they can struggle on multiple-choice questions that involve subjective and nuanced judgments with implicit standards. We suggest using LLMs as an alternative strategy for human evaluation, but with caution and verification by measuring agreement with human preferences.

Lastly, we proposed a perspective of using LLMs as "annotators" rather than classifiers, harnessing their ability to create abundant data. By utilizing LLMs to generate pseudo labels for unlabeled data, we can distill a lightweight classifier that can be reliably deployed at scale. In our experiments, such a classifier achieved competitive results, and matched or even surpassed the performance of GPT-4 as a classifier. We advocate for a careful assessment of the potential use cases of LLMs, balancing performance and efficiency, while exploiting both their power to generalize with the maturity, speed, and cost of conventional machine learning classifiers.

6 DISCUSSION AND FUTURE WORK

This work has the potential to create significant impact for research and application of AI technologies in text mining. Our framework has demonstrated the ability to use LLMs as taxonomy generators, as well as data labelers and evaluators. These automations could lead to significant efficiency gains and cost savings for a variety of domains and applications that rely on understanding, structuring and analyzing massive volumes of unstructured text. It could also broadly democratize the process of mining knowledge from text, empowering non-expert users and enterprises to interact with and interpret their data through natural language, thereby leading to better insights and data-driven decision making for a range of industries and sectors. Additionally, our framework and research findings relate to other work that leverages LLMs for taxonomy creation and text clustering, and has important empirical lessons for the efficient use of instruction-following models in these scenarios.

Despite these initial successes, there are some important challenges and future directions that are worth exploring. As we have already noted, LLMs are expensive and slow. In future work, we hope to explore ways to improve the speed, efficiency and robustness of our framework, through hybrid approaches that further explore the combination of LLMs with embedding-based methods, or model distillation that fine-tunes a smaller model through instructions from a larger one. Evaluation continues to be a crucial and open challenge for future work, and we plan to explore ways of performing more robust LLM-aided evaluations in future work, for example by fine-tuning a model to expand its reasoning capabilities beyond pairwise judgement tasks. While this work has focused largely on text mining in the conversational domain, we also hope to explore the extensibility of our framework to other domains as well. Finally, many domains have ethical considerations from the perspective of privacy and security that must be taken into account when performing large-scale automated text mining, and we hope to engage with these challenges more deeply in future work.

REFERENCES

[1] Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text clustering algorithms. *Mining text data* (2012), 77–128.

[2] David Arthur, Sergei Vassilvitskii, et al. 2007. k-means++: The advantages of careful seeding. In *Soda*, Vol. 7. 1027–1035.

[3] Léon Bottou. 1998. Online algorithms and stochastic approximations. *Online learning in neural networks* (1998).

[4] B Barla Cambazoglu, Leila Tavakoli, Falk Scholer, Mark Sanderson, and Bruce Croft. 2021. An intent taxonomy for questions asked in web search. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. 85–94.

[5] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems* 22 (2009).

[6] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[7] Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement* 33, 3 (1973), 613–619.

[8] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056* (2023).

[9] Simon Haykin. 1998. *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

[10] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. 2005. A brief survey of text mining. *Journal for Language Technology and Computational Linguistics* 20, 1 (2005), 19–62.

[11] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* (2016).

[12] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* (2016).

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[14] Dong-Ho Lee, Jay Pujara, Mohit Sewak, Ryen White, and Sujay Jauhar. 2023. Making Large Language Models Better Data Creators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 15349–15360. https://doi.org/10.18653/v1/2023.emnlp-main.948

[15] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172* (2023).

[16] Geoffrey J McLachlan and Kaye E Basford. 1988. *Mixture models: Inference and applications to clustering*. Vol. 38. M. Dekker New York.

[17] Chau Minh Pham, Alexander Hoyle, Simeng Sun, and Mohit Iyyer. 2023. TopicGPT: A Prompt-based Topic Modeling Framework. *arXiv preprint arXiv:2311.01449* (2023).

[18] Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic Prompt Optimization with "Gradient Descent" and Beam Search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 7957–7968. https://doi.org/10.18653/v1/2023.emnlp-main.494

[19] Daniel E Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*. 13–19.

[20] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[21] Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia* 6, 12 (2008), e26752.

[22] Chirag Shah, Ryen W White, Reid Andersen, Georg Buscher, Scott Counts, Sarkar Snigdha Sarathi Das, Ali Montazer, Sathish Manivannan, Jennifer Neville, Xiaochuan Ni, et al. 2023. Using large language models to generate, validate, and apply user intent taxonomies. *arXiv preprint arXiv:2309.13063* (2023).

[23] Jingbo Shang, Xinyang Zhang, Liyuan Liu, Sha Li, and Jiawei Han. 2020. Nettaxo: Automated topic taxonomy construction from text-rich network. In *Proceedings of the Web Conference 2020*. 1908–1919.

[24] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[25] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. https://arxiv.org/abs/2212.09741

[26] Ah-Hwee Tan et al. 1999. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 workshop on knowledge discovery from advanced databases*, Vol. 8. 65–70.

[27] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences. *arXiv preprint arXiv:2309.10621* (2023).

[28] Ike Vayansky and Sathish AP Kumar. 2020. A review of topic modeling methods. *Information Systems* 94 (2020), 101582.

[29] Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-Driven Explainable Clustering via Language Descriptions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 10626–10649. https://doi.org/10.18653/v1/2023.emnlp-main.657

[30] Anuradha Welivita and Pearl Pu. 2020. A Taxonomy of Empathetic Response Intents in Human Social Conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*. 4886–4899.

[31] Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2104–2113.

[32] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2701–2709.

[33] Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. Can Large Language Models Transform Computational Social Science? *arXiv preprint arXiv:2305.03514* (2023).

A PROMPT TEMPLATES

```
# Instruction
## Context
- **Goal**: Summarize the input text for the given use case.
- **Data**: Your input data is a conversation history between a User and an AI
  agent.
- **Use case**: {{use_case}}

# Data
{{input_text}}

# Questions
## Q1. Provide a summary of the input text **in {{summary_length}} words or less
   ** for the use case.
Tips
- The summary will represent the input data for clustering in the next step.
- Be concise and clear. Do not add phrases like "This is the summary of the data
   ..." or "Summarized text: ...".
- Within {{summary_length}} words, include the relevant information for the use
  case in the summary as much as possible.
- Do not include any line breaks in the summary.
- Provide your answer in **English** only.

## Q2. Explain how you wrote the summary **in {{explanation_length}} words or
   less**.

## Provide your answers between the tags: <summary>your answer to Q1</summary>,
   <explanation>your answer to Q2</explanation>.

# Output
```

Prompt 1: Conversation summarization prompt (Stage 1 in Phase 1).

```
# Instruction
## Context
- **Goal**: Your goal is to cluster the input data into meaningful categories
  for the given use case.
- **Data**: The input data will be a markdown table with summaries for a list of
  human-AI conversations, including the following columns:
  - **id**: conversation index.
  - **text**: conversation summary.
- **Use case**: {{use_case}}

## Requirements
### Format
- Output clusters as a **markdown table** with each row as a category, with the
  following columns:
  - **id**: category index starting from 1 in an incremental manner.
  - **name**: category name should be **within {{cluster_name_length}} words**.
    It can be either *verb phrase* or *noun phrase*, whichever is more
    appropriate.
  - **description**: category description should be **within {{
    cluster_description_length}} words**.
Here is an example of your output:
```markdown
|id|name|description|
|-|-|-|
|category id|category name|category description|
```
- Total number of categories should be **no more than {{max_num_clusters}}**.
- Output table should be in **English** only.
Quality
- **No overlap or contradiction** among the categories.
```

- **Name** is a concise and clear label for the category. Use only phrases that are specific to each category and avoid those that are common to all categories.
- **Description** differentiates one category from another.
- **Name** and **description** can **accurately** and **consistently** classify new data points **without ambiguity**.
- **Name** and **description** are *consistent with each other*.
- Output clusters match the data as closely as possible, without missing important categories or adding unnecessary ones.
- Output clusters serve the given use case well.
- Output clusters should be specific and meaningful. Do not invent categories that are not in the data.

Data
{{data_table}}

Questions
Q1. Please generate a cluster table from the input data that meets the requirements.
Tips
- The cluster table should be a **flat list** of **mutually exclusive** categories. Sort them based on their semantic relatedness.
- You can have *fewer than {{max_num_clusters}} categories* in the cluster table, but **do not exceed the limit.**
- Be **specific** about each category. **Do not include vague categories** such as "Other", "General", "Unclear", "Miscellaneous" or "Undefined" in the cluster table.
- You can ignore low quality or ambiguous data points.

Q2. Why did you cluster the data the way you did? Explain your reasoning **within {{explanation_length}} words**.

Provide your answers between the tags: <cluster_table>your generated cluster table with no more than {{max_num_clusters}} categories</cluster_table>, <explanation>explanation of your reasoning process within {{explanation_length}} words</explanation>.

Output

Prompt 2: Taxonomy generation prompt (Stage 2 in Phase 1).

Instruction
Context
- **Goal**: You goal is to review the given reference table based on the input data for the specified use case, then update the reference table if needed.
    - You will be given a reference cluster table, which is built on existing data. The reference table will be used to classify new data points.
    - You will compare the input data with the reference table, output a rating score of the quality of the reference table, suggest potential edits, and update the reference table if needed.
- **Reference cluster table**: The input cluster table is a markdown table with each row as a category, with the following columns:
  - **id**: category index.
  - **name**: category name.
  - **description**: category description used to classify data points.
- **Data**: {{same as in the taxonomy generation prompt}}
- **Use case**: {{use_case}}

Requirements
{{same as in the taxonomy generation prompt}}

Reference cluster table
{{cluster_table}}

Data
{{data_table}}

Questions
Q1: Review the given reference table and the input data and provide a rating score of the reference table. The rating score should be an integer between 0 and 100, higher rating score means better quality. You should consider the following factors when rating the reference cluster table:
- **Intrinsic quality**:
  - 1) if the cluster table meets the *Requirements* section, with clear and consistent category names and descriptions, and no overlap or contradiction among the categories;
  - 2) if the categories in the cluster table are relevant to the given use case;
  - 3) if the cluster table includes any vague categories such as "Other", "General", "Unclear", "Miscellaneous" or "Undefined".
- **Extrinsic quality**:
  - 1) if the cluster table can accurately and consistently classify the input data without ambiguity;
  - 2) if there are missing categories in the cluster table but appear in the input data;
  - 3) if there are unnecessary categories in the cluster table that do not appear in the input data.

Q2: Explain your rating score in Q1 **within {{explanation_length}} words**.

Q3: Based on your review, decide if you need to edit the reference table to improve its quality. If yes, suggest potential edits **within {{suggestion_length}} words**. If no, please output "N/A".
Tips:
- You can edit the category name, description, or remove a category. You can also merge or add new categories if needed. Your edits should meet the *Requirements* section.
- The cluster table should be a **flat list** of **mutually exclusive** categories. Sort them based on their semantic relatedness.
- You can have *fewer than {{max_num_clusters}} categories* in the cluster table, but **do not exceed the limit.**
- Be **specific** about each category. **Do not include vague categories** such as "Other", "General", "Unclear", "Miscellaneous" or "Undefined" in the cluster table.
- You can ignore low quality or ambiguous data points.

Q4: If you decide to edit the reference table, please provide your updated reference table. If you decide not to edit the reference table, please output the original reference table.

Provide your answers between the tags: <rating>your answer to Q1 between 0 and 100</rating>, <explanation>your answer to Q2 within {{explanation_length}} words</explanation>, <suggestion>your answer to Q3 within {{suggestion_length}} words</suggestion>, <cluster_table>your answer to Q4 in markdown table format with no more than {{max_num_clusters}} categories</cluster_table>.

Output

Prompt 3: Taxonomy update prompt (Stage 2 in Phase 1).

Instruction
Context
- **Goal**: You goal is to review the given reference table based on the requirements and the specified use case, then update the reference table if needed.
    - You will be given a reference cluster table, which is built on existing data. The reference table will be used to classify new data points.
    - You will compare the reference table with the given requirements, output a rating score of the quality of the reference table, suggest potential edits, and update the reference table if needed.
- **Reference cluster table**: {{same as in the taxonomy update prompt}}
- **Use case**: {{use_case}}

Requirements
{{same as in the taxonomy generation prompt}}

Reference cluster table
{{cluster_table}}

Questions
Q1: Review the given reference table and provide a rating score. The rating score should be an integer between 0 and 100, higher rating score means better quality. You should consider the following factors when rating the reference cluster table:
- **Intrinsic quality**:
  {{same as in the taxonomy update prompt}}

Q2: {{same as in the taxonomy update prompt}}

Q3: {{same as in the taxonomy update prompt}}

Q4: {{same as in the taxonomy update prompt}}

Provide your answers between the tags: {{same as in the taxonomy update prompt}}

Output

Prompt 4: Taxonomy review prompt (Stage 2 in Phase 1).

Instruction
Context
- **Goal**: Your goal is to classify the input data using the provided reference table.
- **Reference table**: The input reference table is a markdown table with each row as a category, with the following columns:
  - **id**: category index.
  - **name**: category name.
  - **description**: category description used to classify data points.
- **Data**: Your input data is a conversation history between a User and an AI agent.

Reference table
{{cluster_table}}

Data
{{input_text}}

Questions
Please classify the input data using the reference table. Your output should include the following information:
- **category-id**: **id** of a category in the reference table; if unable to classify using the reference table, please output "-1".
- **category-name**: **name** of a category in the reference table that corresponds to the **category-id**; if unable to classify using the reference table, please output "Undefined".
- **explanation**: a short explanation of why you think the input data belongs to the category or you cannot classify the data into any of the given categories. You explanation should be within {{explanation_length}} words.
Tips
- You should only output the **primary** category for the input data. If it can be classified into multiple categories, please output **the most relevant category**.
- Your output should be in *English* only.

Please provide your answers between the tags: <category-id>your identified category id</category-id>, <category-name>your identified category name</category-name>, <explanation>your explanation</explanation>.

Output

Prompt 5: Label assignment prompt (Phase 2).

B   TAXONOMIES

Label Name	Label Description
Website Navigation Requests	User seeks to visit a very specific website or web page by providing a URL, or keywords that *indicate the name or domain of the website*, e.g., "amazon.com", "gmail login".
Fact-Based Information Seeking	User seeks factual and descriptive information on a specific topic, product, or service. These user queries can be answered by retrieving the factual information that *already exists in the sources* and require *a high level of specificity* and *low level of subjectivity*, e.g., "What is the capital of France?".
Clarification and Concept Explanation	User asks AI to explain various topics or concepts, or seeks clarification or confirmation on a matter, by providing a question that requires more than a factual or a descriptive answer, but rather *an interpretation, definition, or elaboration*, e.g., "What is the difference between AI and machine learning?".
General Solution and Advice Seeking	User seeks general solutions, advice, instructions, or steps on a **non-technical** topic, product, or service, by providing a problem, goal, or scenario that requires more than a factual or descriptive answer, but rather *a recommendation, suggestion, or guidance*, e.g., "What should I buy for my friend's birthday?".
Technical Assistance and Problem Solving	User seeks help with **technical** issues or problem-solving related to a product, service, or system, by providing a description of the issue, error, or challenge that requires more than a factual or descriptive answer, but rather *a diagnosis, solution, or workaround*, e.g., "How to fix the bug in my code?".
Language Translation Requests	User requests translation or interpretation of a phrase or sentence *from one language to another*, e.g., "'Hello' in Spanish".
Content Creation and Storytelling Requests	User requests the *creation of original content* such as images, stories, instructions, summaries, or narratives on a specific topic or theme, e.g., "Create an image of a unicorn in a forest".
Planning and Scheduling	User seeks assistance with planning an event, trip, or schedule, e.g., "Plan a birthday party for my mom".
Data Analysis and Calculation Requests	User asks for quantitative data analysis, calculations, or statistical interpretations, by providing the source of the data and the desired operation or result, e.g., "Calculate the average of these numbers", "Analyze the sales data for last quarter".
Greetings and Social Interactions	User greets the AI agent or engages in social interactions, by providing a salutation, expression, or remark, or requesting to play games with the AI, which *does not require a factual, descriptive, or technical answer*, but rather an engaging, polite or humorous response, e.g., "Hello, how are you?", "You're very smart".

(a) User intent taxonomy.

Label Name	Label Description
Academic Resources	Requests for educational resources, explanations of *general* academic concepts, and academic advice, e.g., "Best college for computer science", "How to prepare for the SAT?".
Linguistics and Language Learning	Requests for translations, text editing, or discussions about grammar, syntax, and other linguistic concepts, e.g., "Translate 'Hello' to French", "What is the difference between 'affect' and 'effect'?".
Mathematics, Logics and Data Science	Queries and discussions related to concepts, theories, and problems in the fields of mathematics and logics, or related to machine learning and data science, e.g., "How to calculate standard deviation?", "What is the difference between boosting and bagging?".
Physics and Chemistry	Queries and discussions related to concepts, theories, and problems in the fields of physics and chemistry, e.g., "What is the speed of light?", "What is the atomic number of carbon?".
Business and Industry	Discussions about *business operations*, *industry developments*, and related information, e.g., "What is the best business strategy for a startup?", "Generate a FAQ page for a healthcare product website".
Economics and Finance	Discussions about economic concepts and theories, financial products, investment advice, and related queries, e.g., "What is the current inflation rate?", "What is the best investment strategy in 2024?".
Job and Career Advice	Requests for job applications, career advice, and related information, e.g., "What is the best career path for a data scientist?".
Legal and Regulatory Information	Queries about legal terms, regulations, and related information, e.g., "What is the legal drinking age in the US?", "What are the regulations for AI development in EU countries?".
Art, Design and Creativity	Requests for *image creation and creative writing*, or discussions about *art, design and creative concepts*, e.g., "Create a logo for my company", "What is the difference between modern art and contemporary art?".
Entertainment, Media, and Gaming	Discussions about movies, music, games, game development, and other forms of entertainment, e.g., "Who is the director of the movie 'Oppenheimer'?".
Interactive Activities with AI	Requests for playing games, or engaging in *interactive activities with the AI*, e.g., "Play a game with me", "Tell me a joke".
Personal Lifestyle and Hobbies	Conversations about *personal* hobbies, lifestyle choices, and individual interests, e.g., "How to learn to play the guitar as a beginner?".
Sports and Fitness	Conversations about *sports events*, *fitness advice*, and related topics, e.g., "Who will play in the NBA finals?", "Training tips for marathon".
Food and Nutrition	Conversations about food recommendations, nutritional information, and cooking advice., e.g., "How to make a pizza?".
Health and Wellness	Discussions about health conditions, treatments, and wellness information, e.g., "Is cancer curable?", "Best practices to improve sleep quality".
General Digital Support	Conversations related to the AI's abilities, limitations, functionality, task requests, and technical support for *general* digital products or services, e.g., "What can Bing Chat do?", "How to take a screenshot on macbook?".
Software Development and Hardware Issues	Conversations about *coding*, *software configuration*, *development tools*, and specific software or *hardware issues* and their solutions, e.g., "How to install python on macbook?", "How to fix a broken external hard drive?".
Home and Household Issues	Queries about home maintenance, household issues, and related advice, e.g., "How to clean a microwave oven?".
Animals and Nature	Queries about animals, nature, and related information, e.g., "What is the pH value of water?", "What is the average lifespan of a cat?".
Geography, Climate and Environment	Queries about geographical facts, weather conditions, climate information, environmental issues and related topics, e.g., "What is the most populous city in the world?", "What is the weather like in Seattle?".
History and Culture	Queries about historical events, cultural practices, and related topics, e.g., "What is the date of the French Revolution?".
Personal Counseling and Emotional Support	Conversations seeking emotional support, personal relationship advice, and life guidance, e.g., "How to deal with a breakup?".
Social and Political Issues	Conversations about social issues, political events, and related topics, e.g., "Who are the candidates for the 2024 US presidential election?".
Product and Shopping Queries	Requests for product suggestions, comparisons, online shopping, product availability, and other related information about consumer goods, e.g., "What is the best laptop for gaming?", "Compare different models of iPhone".
Travel and Tourism	Queries about travel plans, tourist destinations, travel-oriented cultural tips and related information, e.g., "Plan a 5-day trip to Hawaii", "Best place to visit in Paris".

(b) Conversation domain taxonomy.

Table 5: Taxonomies used in the label assignment experiments. Note all presented examples are artificial and do not link to any particular data point in our corpus.