h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

Hochschule Darmstadt

Fachbereiche Mathematik und Naturwissenschaften & Informatik

Uncovering Fine-Grained Intents in Corporate Email: Methodology, Dataset Characterization, and Label Quality Assessment

Abschlussarbeit zur Erlangung des akademischen Grades

Master of Science (M. Sc.)

im Studiengang Data Science

vorgelegt von

Patrick Deck

Matrikelnummer: 1113069

Referent	:	Prof. Dr. Markus Döhring
Korreferent	:	Prof. Dr. Timo Schürg

Ausgabedatum	:	14.10.2024
Abgabedatum	:	14.04.2025

DECLARATION

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, May 19, 2025

Patrick Deck

ABSTRACT

Understanding fine-grained user intent within corporate email is crucial for workplace automation but faces significant Natural Language Processing (NLP) challenges due to unstructured conversational context, unknown intent taxonomies, and pervasive semantic overlap between related goals. This thesis addresses this complexity by developing, implementing, and evaluating a novel semi-automated workflow for discovering and labeling intents in the large-scale Avocado email corpus. The methodology integrates targeted filtering, weak supervision utilizing Large Language Models (LLMs) (using LLaMA 3 8B Instruct for quality scoring and feature generation), semantic clustering (with SBERT embeddings), and an iterative refinement process. This produced a new labeled dataset of 6,785 utterances across 54 identified fine-grained corporate intent classes.

Characterization of this dataset revealed significant structural properties: a highly skewed distribution (with most defined intents being sparse in the underlying corpus, confirmed via random sampling) and substantial semantic overlap between related classes (low Silhouette Score: 0.057, high DBI: 3.257). Comparative analysis positioned its structural complexity nearest the BANKING77 benchmark, highlighting challenges related to shared vocabulary for nuanced intents, distinct from benchmarks benefiting from greater topical diversity or strong keywords.

Label quality assessment using Confident Learning (Cleanlab) flagged 4.2% of labels as potential inconsistencies. Further analysis indicated these flagged instances primarily highlight the inherent difficulties of assigning definitive single labels in this domain, frequently occurring where utterances reflect semantic ambiguity or fall on inherently fuzzy boundaries between subtle intent categories. This suggests the identified inconsistencies reflect the complexity of mapping nuanced communication onto a discrete taxonomy.

In conclusion, this work contributes: (1) An adaptable workflow demonstrating LLM utility and limitations as weak supervisors in intent discovery; (2) A unique, characterized dataset embodying real-world email complexities like overlap and ambiguity; and (3) Insights from label quality assessment diagnosing inconsistencies tied to inherent domain characteristics. A key finding underscored by this assessment is that even when attempting to simplify the problem by focusing on individual sentences, the prevalence of multi-intent expressions often makes a single label insufficient for accurately capturing the full communicative meaning. The findings collectively emphasize the need for context-aware, potentially multi-label approaches to achieve deeper communication understanding in complex domains like corporate email.

keywords: Intent Discovery, Corporate Email, Fine-Grained Intents, Natural Language Processing, Dataset Creation, Labeling Workflow, Large Language Models

ZUSAMMENFASSUNG

Das Verstehen von feingranularen Nutzerintentionen innerhalb von Unternehmens-E-Mails ist für die Automatisierung des Arbeitsumfelds von entscheidender Bedeutung, stellt jedoch aufgrund des unstrukturierten Konversationskontexts, unbekannter Intentionstaxonomien und weitreichender semantischer Überlappungen zwischen verwandten Zielen eine erhebliche Herausforderung für die natürliche Sprachverarbeitung (NLP) dar. Diese Arbeit befasst sich mit dieser Komplexität durch die Entwicklung, Implementierung und Evaluierung eines neuartigen semi-automatischen Workflows zur Erkennung und Labeling von Intentionen im umfangreichen Avocado E-Mail-Korpus. Die Methodik integriert gezielte Filterung, schwache Überwachung unter Verwendung von Large Language Models (LLMs) (unter Verwendung von LLaMA 3 8B Instruct für Qualitätsbewertung und Feature-Generierung), semantisches Clustering (mit SBERT Embeddings) und einen iterativen Verfeinerungsprozess. Auf diese Weise entstand ein neuer gelabelter Datensatz mit 6.785 Äußerungen über 54 identifizierte, feingranulare Klassen von Intentionen aus dem Unternehmensumfeld.

Die Charakterisierung dieses Datensatzes offenbarte signifikante strukturelle Eigenschaften: eine stark verzerrte Verteilung (im zugrunde liegenden Korpus sind die meisten definierten Intentionen nur selten zu finden, was durch Stichproben bestätigt wurde) und erhebliche semantische Überlappungen zwischen verwandten Klassen (niedriger Silhouette Score: 0,057; hoher DBI: 3,257). Eine vergleichende Analyse ergab, dass die Struktur am nächsten an dem BANKING77-Benchmark liegt, was die Herausforderungen im Zusammenhang mit dem geteilten Vokabular für nuancierte Intentionen hervorhebt, im Gegensatz zu Benchmarks, die von einer größeren thematischen Vielfalt oder starken Schlagwörtern profitieren.

Bei der Bewertung der Label-Qualität mit Confident Learning (Cleanlab) wurden 4,2% der Labels als potenzielle Unstimmigkeiten eingestuft. Weitere Analysen ergaben, dass diese markierten Instanzen in erster Linie die inhärenten Schwierigkeiten bei der Zuweisung definitiver Single-Labels in diesem Bereich verdeutlichen. Sie treten häufig dort auf, wo Äußerungen semantische Mehrdeutigkeit widerspiegeln oder auf unscharfe Grenzen zwischen subtilen Intentionskategorien fallen. Dies deutet darauf hin, dass die festgestellten Unstimmigkeiten die Komplexität der Zuordnung nuancierter Kommunikation zu einer diskreten Taxonomie widerspiegeln.

Zusammenfassend trägt diese Arbeit zu Folgendem bei: (1) Einem anpassungsfähigen Workflow, der den Nutzen und die Grenzen von LLMs als schwache Supervisoren bei der Entdeckung von Intentionen aufzeigt; (2) einem einzigartigen, charakterisierten Datensatz, der die Komplexität von E-Mails in der realen Welt verkörpert, wie z.B. Überlappungen und Mehrdeutigkeit; und (3) Erkenntnissen aus der Bewertung der Label-Qualität, die Unstimmigkeiten in Verbindung mit inhärenten Domänenmerkmalen diagnostiziert. Eine wichtige Erkenntnis, die durch diese Auswertung unterstrichen wird, ist, dass selbst bei dem Versuch, das Problem zu vereinfachen, indem der Fokus

auf einzelne Sätze gelegt wurde, die Prävalenz von Ausdrücken mit mehreren Intentionen oft dazu führt, dass ein einzelnes Label nicht ausreicht, um die ganzheitliche kommunikative Bedeutung exakt zu erfassen. Die Ergebnisse unterstreichen den Bedarf an kontextbezogenen, potenziell Multi-Label-Ansätzen, um ein tieferes Verständnis der Kommunikation in komplexen Domänen wie Firmen-E-Mails zu erreichen.

Schlagwörter: Intent Discovery, Corporate Email, Fine-Grained Intents, Natural Language Processing, Dataset Creation, Labeling Workflow, Large Language Models

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

LIST OF ALGORITHMS

LIST OF ACRONYMS

ADB Adaptive Decision Boundary.
API Application Programming Interface.
ARI Adjusted Rand Index.

BERT Bidirectional Encoder Representations from Transformers.
BoW Bag of Words.

CC Contrastive Clustering.
CDAC+ Constrained Deep Adaptive Clustering with Cluster Refinement.
CV Computer Vision.

DBI Davies-Bouldin Index.
DNN Deep Neural Network.

ELMo Embeddings from Language Models.

GloVe Global Vectors.
GMM Gaussian Mixture Model.
GPT Generative Pre-Trained Transformer.

IAA Inter-Annotator Agreement.
ICL In-Context Learning.
IDF Inverse Document Frequency.

KIR Known Intent Ratio.

LLM Large Language Model.
LMCL Large Margin Cosine Loss.
LOF Local Outlier Factor.
LR Labeled Ratio.
LSTM Long Short-Term Memory.

NLP Natural Language Processing.
NMI Normalized Mutual Information.

OOD Out-of-Domain.
OOS Out-of-Sample.
OOV Out-of-Vocabulary.
OSR Open Set Recognition.

PST Personal Storage Table.

RNN Recurrent Neural Network.

SBERT Sentence BERT.
SCCL Supporting Clustering with Contrastive Learning.
SEG Semantic-Enhanced Gaussian Mixture Model.

T5 Text-to-Text Transfer Transformer.
TEXTOIR Text Open Intent Recognition.
TF Term Frequency.
TF-IDF Term Frequency - Inverse Document Frequency.

UMAP Uniform Manifold Approximation and Projection for Dimension Reduction.
USNID Unsupervised and Semi-Supervised New Intent Discovery.

Part I

THESIS

1 INTRODUCTION

Understanding the underlying intent in textual communication is a fundamental challenge in Natural Language Processing (NLP). In corporate environments, email remains a primary mode of communication, facilitating task coordination, decision-making, and information exchange. Unlike structured interactions in chatbots or customer support logs, where each message typically revolves around a single request, emails serve a much broader range of functions within an ongoing conversational context. They can involve requests for action or information, provide information, or serve other means. Moreover, it's not unusual for emails to carry multiple intents at once, often involving subtle distinctions and significant semantic overlap between related communication goals, making the process of uncovering underlying, fine-grained intents even more complex.

Traditional intent classification relies on a fixed set of pre-defined intent categories. However, for real-world scenarios like diverse corporate email, the relevant intent taxonomy is often unknown and needs to be uncovered. This makes intent discovery—the process of identifying new and previously unseen intent categories—an essential aspect of understanding communication patterns in this domain. While conceptually related to clustering, the nature of corporate email, focused on broad communication rather than just explicit requests, means that simple clustering algorithms often fail to yield meaningful intent categories. A specialized approach is necessary.

Recent advances in machine learning, particularly deep learning and Large Language Models (LLMs), offer sophisticated methods for text analysis. However, their application to fine-grained intent discovery specifically within corporate emails remains relatively unexplored. Furthermore, manual annotation is time-consuming and requires domain expertise, while automated methods demand careful validation to ensure data quality, especially given the inherent ambiguities.

To address these challenges, this thesis introduces and evaluates a novel, data-driven workflow involving targeted data filtering, LLM-based feature generation, semantic clustering, and an iterative refinement process for discovering and labeling fine-grained intents in corporate emails. The methodology aims to provide a scalable solution for analyzing complex email data. In doing so, this research contributes not only the workflow and the resulting labeled dataset but also a detailed characterization of this dataset's unique structural properties compared to standard benchmarks, and a critical assessment of the achieved label quality, offering insights into the practical challenges of the task and advancing the understanding of fine-grained intent discovery in emails.

The remainder of this chapter is structured as follows: Section 1.1 presents the motivation for studying intent detection and discovery in corporate emails. Section 1.2 outlines the research questions that guide this work. Finally, Section 1.3 outlines the organization of the thesis.

1.1 MOTIVATION

Corporate email communication remains a vital medium for workplace coordination, decision-making, and knowledge exchange. Despite its significance, the underlying intents within these emails remain largely unstructured and difficult to analyze systematically. Understanding the specific goal behind an email message – whether it involves a request for action, an approval, an inquiry for specific information, or an offer of assistance – provides valuable insights into communication patterns. These insights can enable organizations to streamline processes, reduce inefficiencies, enhance decision-making, and ultimately improve productivity.

While understanding the general purpose of an email is useful, significant potential for sophisticated workflow automation and communication analytics lies in identifying more specific user goals. This thesis, therefore, focuses on uncovering and analyzing **fine-grained intents** within corporate email. We define "fine-grained intents" as the specific, often subtle, communicative goals or desired actions expressed typically at the sentence or key phrase level. This requires distinguishing between closely related functions – for example, differentiating a request to *set up* a meeting from a request to *reschedule* one, or distinguishing *requesting a document copy* from *requesting to be copied (CC'd) on an email*. Capturing this level of detail is crucial for building truly helpful automated assistants and understanding nuanced communication patterns, but it presents considerable challenges due to the inherent ambiguity, context-dependence, and semantic overlap prevalent in real-world email conversations.

Unlike many traditional intent classification tasks focused on direct user commands (e.g., in chatbots), corporate emails present unique difficulties. The full range of possible intents is unknown beforehand, necessitating an exploratory, data-driven approach to intent discovery. Furthermore, workplace communication is highly dynamic, influenced by company culture, specific jargon, and evolving workflows, further complicating the identification and consistent *labeling* of these fine-grained intent categories.

Once potential fine-grained intents are identified, structuring them into a high-quality labeled dataset is a critical, yet challenging, next step for enabling model training and evaluation. However, publicly available benchmark datasets that specifically combine fine-grained intent annotations with the diverse, conversational context of corporate email are largely lacking. While established corporate email corpora like the Enron dataset [KY04] exist, they typically lack the necessary granular intent labels required for this type of analysis. Conversely, standard intent classification benchmarks (e.g., SNIPS [Cou+18], BANKING77 [Cas+20]) originate from different domains, such as task-oriented dialogue or specific customer service interactions, and do not capture the unique characteristics and communication patterns inherent in corporate email exchanges.

This gap in readily available, suitably annotated resources makes it difficult to develop, train, and rigorously evaluate NLP models specifically designed for understanding nuanced intentions within the corporate email domain. It also hinders direct comparison and benchmarking of different approaches. Consequently, a key prerequisite for

advancing research in this area involves the creation and characterization of datasets that begin to capture these fine-grained distinctions and associated challenges.

Given this lack of suitable public datasets and the inherent difficulties in identification and categorization, intent labeling in this context requires more than just a technical solution; it necessitates a systematic workflow to ensure consistency and reproducibility, moving beyond purely manual, labor-intensive efforts which are often subjective and infeasible for large datasets.

This research, therefore, contributes not only a novel labeled dataset reflecting the complexities of fine-grained corporate email intents but also a methodological framework for discovering and categorizing these intents. By bridging the gap between exploratory intent discovery and structured dataset creation, this study aims to support advancements in email data analysis, workplace communication insights, and AI-driven automation in corporate settings.

1.2 RESEARCH QUESTIONS

This thesis addresses the challenge of understanding user intent within large corporate email datasets. It focuses on developing and evaluating methodologies for discovering, labeling, and characterizing fine-grained intents in this complex domain. The research is guided by the following key questions:

1. **RQ 1:** How can fine-grained user intents be effectively identified, categorized, and labeled at scale within a corporate email corpus, and what role can LLMs play in facilitating this complex task?

2. **RQ 2:** What are the structural characteristics (intent distribution, semantic separability, lexical patterns) of a labeled dataset representing fine-grained corporate email intents, and how do these characteristics compare to established intent benchmark datasets?

3. **RQ 3:** Given the inherent challenges of semantic ambiguity and potential multi-intent expressions in corporate email, what systematic methods can be employed to assess the quality and consistency of labels in a dataset representing fine-grained intents from this domain?

1.3 THESIS STRUCTURE

The remainder of this thesis is organized as follows:

- **Chapter 2: Background**
  Chapter 2 provides the necessary foundation for understanding the research context. It begins with a historical overview of text embeddings and representations, tracing the evolution from early approaches to modern deep learning-based techniques. The chapter then covers relevant work in the broader fields of intent detection and intent discovery, as well as the increasing use of LLMs for data

annotation tasks. Subsequently, it reviews prior research focused specifically on analyzing communication patterns and identifying intents or speech acts within email data, noting that such analyses have typically operated at a coarser level of granularity than the fine-grained focus of this thesis. Finally, standard benchmark datasets commonly used in intent analysis are introduced to provide context for later comparisons.

- **Chapter 3: Methodology**
  Chapter 3 details the systematic methodology developed and employed in this research to address RQ1. It describes the data source and necessary preprocessing steps. It then outlines the proposed multi-stage workflow for discovering and labeling fine-grained intents, detailing the use of LLMs for annotation generation, embedding techniques for semantic representation, clustering algorithms for grouping intents, and the iterative process designed for refinement. Finally, this chapter defines the evaluation strategy, outlining the metrics and rationale for assessing dataset characteristics and label quality.

- **Chapter 4: Experimental Setup**
  Chapter 4 presents the specific experimental setup used to implement the methodology and enable the analyses reported in Chapter 5. It defines the final labeled dataset artifact generated by the workflow, including its splits, and lists the benchmark datasets used for comparison. Crucially, it details the procedures followed for: characterizing the labeled dataset (distribution, semantic structure, lexical analysis, quantitative metrics), performing the comparative analysis against benchmarks, conducting the supplementary random sampling analysis, and executing the label quality assessment using Cleanlab, including the cross-validation setup.

- **Chapter 5: Results**
  Chapter 5 presents the empirical findings resulting from the experiments detailed in Chapter 4. This includes a detailed characterization of the generated labeled dataset (qualitative overview, distribution, semantic structure via Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP), lexical patterns, quantitative metrics like Silhouette/Davies-Bouldin Index (DBI)/similarity). It also presents the results of the same characterization analyses applied to the benchmark datasets and a comparative summary of key structural metrics. Furthermore, the results of the supplementary random sampling analysis approximating underlying distributions are presented, alongside the findings from the Cleanlab label quality assessment, highlighting identified inconsistencies.

- **Chapter 6: Discussion**
  Chapter 6 provides an in-depth interpretation and discussion of the results presented in Chapter 5. It synthesizes the findings to address the research questions (RQ1, RQ2, RQ3) posed in Chapter 1. The chapter analyzes the effectiveness and limitations of the developed methodology, discusses the characteristics and quality of the generated dataset in the context of benchmarks and domain challenges (like

semantic overlap and multi-intent utterances), explores the implications of the findings, and acknowledges the study's limitations.

- **Chapter 7: Conclusion and Future Work**
  The final chapter summarizes the entire research effort. It restates the core problem and objectives, highlights the key contributions and main conclusions drawn from the research in response to the guiding questions, and acknowledges limitations. Based on these, it proposes specific directions for future work aimed at improving intent discovery workflows, dataset quality, and the analysis of communication data.

2

BACKGROUND

In this chapter, we provide a comprehensive background on the key concepts and methodologies that underpin this research. We begin with a historical overview of text embeddings and representations, highlighting the challenges associated with encoding textual data and tracing the evolution from early techniques to modern text representations.

The chapter then transitions to the topics of intent detection and discovery, which serve as the foundational concepts for this thesis in identifying fine-grained intents.

Next, we explore relevant work on intent classification in email data, drawing parallels to our own approach.

Finally, we examine the growing role of LLMs in automating data annotation, focusing on their application in generating and evaluating labels for tasks like intent detection.

2.1 HISTORICAL OVERVIEW OF TEXT EMBEDDINGS AND REPRESENTATIONS

Unlike numerical data, which exists on a continuous scale, textual data is discrete and unstructured, making it difficult to process directly. Words are distinct symbols with no natural interpolation, and word order matters, meaning small rearrangements can completely alter meaning. Text also suffers from ambiguity, where words and phrases take on different meanings depending on context. Morphology and grammar further complicate processing, as words change form based on tense and structure, and complex syntax affects interpretation.

To address these challenges, researchers have developed numerical representations of text, such as word embeddings, which capture structure, semantics, and contextual nuances to improve machine understanding of language.

2.1.1 *Early Approaches For Encoding Text as Numerical Representations*

One of the first numerical representations of text was **One-Hot Encoding** and has been used for early rule-based machine translation. The idea is to create a vocabulary of all the unique words in a corpus. Each word is being assigned a specific index for representation. For example under the assumption that a corpus consists of the words `{"cat", "dog", "fish"}` the resulting representation for the word "cat" would be `[1, 0, 0]`. This approach creates sparse (contains a lot of zeros) high dimensional binary vectors with each vector being the size of the whole corpus.

Another early approach for encoding text into a numerical representation is the Bag of Words (BoW) approach. Instead of encoding individual words, the idea of BoW is to encode whole sentences or documents based on word frequency. For a corpus

containing the words `{"I", "like", "cat", "dog", "fish"}` a document containing "I like cat" a corresponding representation would be `[1, 1, 1, 0, 0]`. However, whereas the One-Hot Encoding approach only contains binary values, the BoW approach can contain values greater than one, as it represents the frequency of how often a word occurred in a given document.

A similar approach based on word frequency is Term Frequency - Inverse Document Frequency (TF-IDF). TF-IDF is a statistical measure used to assess the importance of a word within a document relative to a larger corpus. The underlying assumption is that the more frequently a word appears in a document (Term Frequency (TF)), the more significant it is likely to be for that document. However, to counteract the effect of words that appear frequently across many documents (such as common stop words like "the", "is", and "and"), the Inverse Document Frequency (IDF) is introduced. The IDF reduces the weight of words that are very common across the corpus, ensuring that only words that are both frequent in a specific document and rare across the corpus are considered significant. The resulting TF-IDF score assigns higher values to words that are frequent in a document but rare in other documents within the corpus. This helps identify terms that are particularly representative of a document's content, improving the ability to capture distinctive features and thus better distinguishing the document from others.

These early approaches, however, had several significant limitations. The high-dimensional vector representations used in One-Hot Encoding and BoW were inefficient in terms of both memory and computation. As the number of words in the vocabulary grows, the vector spaces become exponentially larger, leading to the curse of dimensionality. This means that the number of possible word combinations increases rapidly, making it harder to capture meaningful patterns and relationships between words. Moreover, these methods failed to capture semantic relationships, as words like "dog" and "cat" were represented as entirely distinct vectors, despite their semantic similarity (e.g., both are animals). Additionally, contextual information was disregarded, since words like "bank" were treated as identical, even though one could refer to a financial institution and the other to the side of a river. Furthermore, word order was neglected, meaning syntactic relationships between words were ignored. These limitations highlighted the need for more advanced methods that could overcome the curse of dimensionality, better capture semantic meaning, and incorporate contextual relationships in a more efficient way.

2.1.2 *Deep Learning Approaches*

To address some of these challenges, the work of Bengio et al. [Ben+03] presents a significant milestone in the development of numerical representations for text. Previous approaches relied on n-grams, short word sequences, to model language by predicting the probability of a word occurring given a preceding sequence. However, this approach struggles to generalize because the number of possible word sequences grows exponentially as the vocabulary size increases. The authors propose a **neural network-based** approach in which each word is mapped to a dense, **continuous-valued feature vector**, allowing the model to capture semantic similarities between words. The core idea is to

train the network to predict the most likely word given a specific context. Through this process, the model learns to create useful, dense representations of words that capture their meanings and is able to generalize beyond training data.

Mikolov et al. [Mik+13] build upon the foundational work by Bengio et al. by introducing **Word2Vec**, making several key improvements. Instead of calculating the probability distribution over all possible words in the vocabulary, Word2Vec employs **negative sampling**, which reduces computational cost by updating only a small subset of words. Additionally, it introduces **hierarchical softmax**, which replaces the expensive full softmax computation with a binary tree structure, reducing complexity from $O(V)$ to $O(\log V)$, with V being the vocabulary. These optimizations make training significantly more efficient while preserving high-quality word representations. Additionally, they found that they can perform simple algebraic operations on their word vectors to derive meaningful results. The operation `vector("King") - vector("Man") + vector("Woman)` results in a vector that is closest to the vector representation of the word `"Queen"`.

In 2014, Pennington et al. [PSM14] proposed Global Vectors (GloVe) as an improvement over Word2Vec. While Word2Vec relies on local context windows to learn word embeddings, GloVe incorporates global corpus statistics by constructing a **word co-occurrence matrix**, where each entry represents how often two words appear together in a given context. Instead of learning embeddings through predictive tasks (as in Word2Vec), GloVe derives them via matrix factorization of this co-occurrence data. This approach captures semantic relationships more effectively by leveraging information from the entire corpus. Additionally, GloVe handles rare words better than Word2Vec, which often struggles to learn meaningful representations for infrequent terms.

Building upon Word2Vec, Bojanowski et al. [Boj+17] propose **FastText**. FastText improves upon previous attempts by incorporating subword information. Instead of treating each word as single units, FastText breaks words into **character n-grams** (subword units). The embedding for a word is constructed by summing over the embeddings calculated for these subword units. This approach handles Out-of-Vocabulary (OOV) words better and is well suited for languages that contain a lot of morphology (e.g., German, Turkish, Finnish). Moreover, words with common roots such as "run", "runner", "running" share subword components, leading to more meaningful embeddings.

2.1.3 *Contextualized Word Embeddings*

While early models captured semantic relationships between words, they lacked contextual awareness—each word had the same embedding regardless of the sentence it appeared in.

One of the first models to address this was Embeddings from Language Models (ELMo), proposed by Peters et al. [Pet+18]. Unlike static word embeddings, ELMo generates **contextualized embeddings** that vary depending on the surrounding text. It uses a bi-directional Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN) designed to process sequential data more effectively than standard RNNs. The bi-directional LSTM processes text in both left-to-right and right-to-left directions, en-

abling ELMo to capture richer contextual information. However, despite improvements over static embeddings, LSTMs still struggle with long-range dependencies, motivating the development of the Transformer architecture.

With the introduction of the **Transformer architecture** by Vaswani et al. [Vas17], a revolutionary shift in NLP occurred. Unlike RNN-based models, Transformers rely on **self-attention mechanisms**, which enable them to weigh the importance of each word relative to others in a sentence. This allows them to effectively capture **long-range dependencies**, a limitation of RNNs and LSTMs. Additionally, Transformers eliminate sequential processing constraints, enabling parallelization, which improves efficiency and scalability. The Transformer consists of two main components: an **Encoder**, which generates contextualized representations of input text, and a **Decoder**, which uses these representations to generate output text. Different Transformer-based models utilize these components in varying ways.

Bidirectional Encoder Representations from Transformers (BERT), introduced by Devlin et al. [Dev+19], consists only of an Encoder. This design reflects BERT's focus on generating deep contextual embeddings rather than producing output text. By leveraging self-attention, BERT considers both left and right context simultaneously, allowing it to better capture word relationships.

However, while BERT produces strong contextualized embeddings, it is not optimized for efficient sentence-level comparisons. A direct application of cosine similarity on BERT embeddings often yields suboptimal results, as BERT is trained for token-level understanding rather than sentence-level meaning. Sentence BERT (SBERT), introduced by Reimers and Gurevych [RG19], addresses this by fine-tuning BERT with a Siamese network structure, enabling it to generate fixed-size sentence embeddings that can be efficiently compared. SBERT significantly improves performance in tasks like semantic search, sentence clustering, and retrieval, where computing similarity between entire sentences is crucial.

Generative Pre-Trained Transformer (GPT) [Ope+24] follows a different approach by using only the Decoder. Unlike BERT, which learns bidirectional representations for contextual understanding, GPT is designed for **autoregressive text generation**—predicting words sequentially based on prior context. This makes it particularly effective for open-ended text generation tasks.

Seq2Seq models such as Text-to-Text Transfer Transformer (T5) [Raf+20] incorporate both an Encoder and a Decoder. The Encoder processes input text to create contextualized representations, which the Decoder then uses to generate meaningful output. This design allows T5 to handle a wide range of NLP tasks by treating them as text-to-text transformations.

As Transformer-based architectures evolved, LLMs emerged, trained on massive text corpora to develop broad knowledge and reasoning abilities. Notable examples include ChatGPT, Gemini [Tea+24], and LLaMA [Gra+24]. Unlike early Transformer-based models, which primarily generated text without specific behavior control, modern LLMs are **fine-tuned for instruction-following**. This enables them not only to generate fluent text but also to follow structured prompts, making them valuable tools for text analysis, transformation, and generation.

While most LLMs are proprietary and accessible only via Application Programming Interface (API) requests, **LLaMA** provides an open-weight alternative, offering several key advantages:

- **Fine-tuning flexibility:** Researchers can adapt LLaMA to specific domains and applications by training it on specialized datasets.
- **Data privacy:** Unlike API-based models, which require sending sensitive data to external servers, LLaMA enables local deployment, ensuring data remains in-house.
- **Accessibility:** By making model weights publicly available, LLaMA allows for experimentation and independent research without relying on proprietary services.

2.1.4 *Cosine Similarity For Embedding comparison*

Once textual utterances are transformed into numerical vector representations (embeddings) using techniques like SBERT (as described in Section 2.1.3), a method is needed to quantify their semantic relatedness. Cosine similarity is a widely adopted metric for this purpose, measuring the similarity between two non-zero vectors in a high-dimensional space based on the cosine of the angle between them.

Formally, the cosine similarity between two vectors A and B is defined as their dot product divided by the product of their magnitudes (or norms):

$$\text{sim}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B are two vectors. The dot product is computed as:

$$A \cdot B = \sum_{i=1}^{n} A_i \cdot B_i$$

and the norm of a vector A is given by:

$$\|A\| = \sqrt{\sum_{i=1}^{n} A_i^2}$$

The resulting similarity score ranges from -1 to 1:

- A score of 1 indicates that the vectors point in the exact same direction (angle of 0°), implying maximum semantic similarity between the corresponding text utterances.
- A score of 0 indicates that the vectors are orthogonal (angle of 90°), suggesting no semantic relationship (dissimilarity).

- A score of -1 indicates that the vectors point in diametrically opposite directions (angle of 180°), implying opposite meanings (though this interpretation is less common for standard sentence embeddings).

APPLICATION TO TEXT EMBEDDINGS. In the context of NLP, each sentence or document embedding (like those generated by SBERT) is a vector in a high-dimensional semantic space. The fundamental idea is that the *direction* of this vector captures the semantic meaning of the text. Cosine similarity, by measuring the angle between two such vectors, effectively compares their directions. Therefore, a high cosine similarity between the embeddings of two sentences suggests they convey similar meanings, regardless of their exact word choice or length.

WHY USE COSINE SIMILARITY? While other distance metrics like Euclidian distance exist, cosine similarity is often preferred for comparing high-dimensional text embeddings due to several key advantages. Firstly, cosine similarity focuses solely on the orientation (direction) of the vectors, not their magnitude. In text embeddings, the direction primarily encodes semantic meaning, while magnitude can sometimes be influenced by factors like sentence length or frequency statistics that may not be relevant for semantic comparison. Euclidian distance, in contrast, is sensitive to magnitude. Secondly, Euclidian distances can become less meaningful in very high-dimensional spaces (the "curse of dimensionality"), where distances between points tend to concentrate. Cosine similarity, focusing on angles, often remains a more robust measure of relatedness in such spaces.

Cosine similarity thus forms the basis for several analyses in this work and is used directly to compare utterance embeddings for tasks like semantic clustering (where items with high similarity are grouped) and similarity search. Furthermore, related metrics like cosine distance (defined as $1 - \textit{cosine\_similarity}$) are employed as the distance measure within algorithms like Agglomerative Clustering and for calculating cluster quality metrics in the forthcoming chapters.

2.1.5 *Comparison of Text Representation Techniques*

The evolution of text representations, as discussed in the preceding sections, moved from simple frequency-based methods to complex, context-aware deep learning models. Table 2.1 summarizes the key characteristics, advantages, and limitations of these approaches.

2.2 RELEVANT WORK

Closely related fields are the fields of intent detection and intent discovery, as well as the field of LLM prompt engineering and similar work that analyzed email data in terms of their intents.

Table 2.1: Comparison of different text representation techniques over time.

Approach/Model	Repr. Level	Vector Type	Core Idea / Method	Context Handling	Key Advantage(s)	Key Limitation(s)
One-Hot Encoding	Word	Sparse Binary	Unique index per word	None	Simple concept	High dim, Sparsity, No semantics, No context
BoW	Doc/Sent	Sparse Count	Word Frequency	None	Simple frequency rep.	High dim, Sparsity, No semantics, No context, No order
TF-IDF	Doc/Sent	Sparse Weighted	Term Freq. + Inverse Doc Freq.	None	Weights important terms	No semantics, No context, No order
Word2Vec	Word	Dense Cont.	Predict context/word (Shallow Network); Neg Samp/Hier SM	Local Window	Efficient, Good semantics (analogies)	Static reps (no context), Struggles w/ rare words
GloVe	Word	Dense Cont.	Factorize Global Co-occurrence Matrix	Global (via matrix)	Good semantics, Better rare words	Static reps (no context)
FastText	Word (Sub)	Dense Cont.	Sum of Character N-gram Embeddings	Subword level	Handles OOV, Morphology aware	Static word reps (no context)
ELMo	Word	Dense Contextual	Bi-LSTM Layers	Full Sentence (Bi-LSTM)	First major contextual model	Struggles w/ long-range dependencies
BERT	Token	Dense Contextual	Transformer Encoder; Masked LM	Full Sequence (Bidirect. Attn)	Deep bidirectional context	Not optimized for sentence similarity
SBERT	Sentence	Dense Fixed-Size	Fine-tune BERT (Siamese Structure)	Full Sentence	Efficient sentence comparison, Semantic search	Requires fine-tuning
GPT	Token	Dense Contextual	Transformer Decoder; Autoregressive Prediction	Left Context only	Strong text generation	Primarily unidirectional understanding
Seq2Seq (T5)	Sequence	Dense Contextual	Transformer Encoder-Decoder	Full Sequence (Bidirect. Enc)	Flexible text-to-text tasks	Complex architecture
LLMs (General)	Sequence	Dense Contextual	Large-Scale Transformer; Instruction Tuning	Full Sequence	Broad knowledge, Instruction following, Reasoning	Computationally expensive, Often proprietary (except LLaMA)

2.2.1 *Intent Detection*

Traditional approaches of intent detection view it as a text classification problem in which the goal is to find the correct mapping of a set of utterances $U = \{u_1, u_2, \ldots, u_n\}$ to a set of classes $C = \{c_1, c_2, \ldots, c_m\}$. This problem can be extended to the problem of *slot filling*, where the task not only consists of correctly identifying the user intent, but also to extract important information that is mandatory to help the user with fulfilling a request. For example for the utterance: *I want to book a flight to New York.* The virtual assistant not only needs to correctly identify the intent (e.g. *book_flight*), but also the requested destination (*New York*). The joint problem of intent detection alongside slot filling, [Zha+18], [E+19], [Qin+19], has been studied extensively and reached impressive results on benchmark datasets such as **ATIS** [HGD90] and **SNIPS** [Cou+18].

Due to the simplicity of the datasets, more challenging datasets have been proposed such as **BANKING77** [Cas+20], which contains 77 fine-grained intent categories from the banking and finance domain. It covers a wide range of topics such as transactions, card issues, account management and security concerncs. The dataset is not only challenging because of the large amount of intent categories, but also because of the semantic overlap between some of the categories. Another more complex dataset is **CLINC150** [Lar+19], which contains 150 intent categories from 10 domains. A comprehensive overview of benchmark datasets commonly used in the fields of intent detection and discovery is presented in a later section: Section 2.2.3.

Even though the complexity of the used benchmark dataset has increased, another problem remained: the proposed approaches for intent detection operate under the assumption of a closed-world, i.e. the number of intents is known and fix. However, this approach assumes that all test classes are known at training time. This does not align with a real-world scenario, where not all user intents are known in advance and may evolve over time. A similar problem exists in the field of Computer Vision (CV) called Open Set Recognition (OSR) [Sch+13]. Fei et al. [FL16] stress the necessity of text classification algorithms to be adapted to a similar problem. Therefore they propose to design classification algorithms in a way to classify documents of the known class into their respective known class, while simultaneously assigning documents, for which the model cannot confidently classify as one of the known classes, as an additional $(m+1)^{th}$ unknown or open (world) class. There is still ongoing research happening using the closed-world assumption, which makes sense for some applications. In this paper however, the focus is on research using the open-world assumption.

With the rise of Deep Neural Networks (DNNs), Bendale et al. [BB16] propose a new approach achieving state-of-the-art performance in OSR by replacing the softmax layer with a distance-aware alternative called **OpenMax**. Instead of directly applying SoftMax to the activations from the penultimate layer, the distance between an input's activations and the mean activations of known classes is measured. If the input is far from all known class distributions, it is classified as unknown; otherwise, it is assigned to the closest known class.

Shu et al. [SXL17] build upon this work and apply it to text document classification. Instead of using a traditional softmax layer, they employ a **1-vs-rest output layer**.

The softmax layer produces a probability distribution over mutually exclusive classes, meaning that every input is always assigned to one of the known classes—even when it does not actually belong to any of them. This makes it difficult to reject unknown samples. In contrast, the 1-vs-rest layer applies independent sigmoid activation functions to each class, allowing the model to assign low confidence scores across all classes when a sample does not fit any known category.

Lin et al. [LX19] propose a two-stage approach: first, a bidirectional LSTM is used as a feature extractor to learn high-level semantic representations of intents. They argue that traditional softmax loss only ensures correct classification but does not enforce intra-class compactness or inter-class separation, which is crucial for distinguishing unknown intents. To address this, they replace the softmax loss with Large Margin Cosine Loss (LMCL), a type of margin loss that applies L_2 normalization to features and weight vectors and introduces a cosine margin to maximize inter-class variance and minimize intra-class variance. In the second stage, the discriminative deep features are fed to a Local Outlier Factor (LOF) algorithm, which evaluates the local density of the features to determine whether a sample belongs to an unknown class. The method was evaluated on benchmark dialogue datasets (SNIPS and ATIS) and demonstrated significant improvements over baseline methods in detecting unknown intents.

Fan et al. [Fan+20] acknowledge the effectiveness of the previously described LMCL approach but highlight two key limitations: (1) LMCL ignores prior knowledge of class labels by focusing solely on maximizing the cosine margin between embeddings, without incorporating semantic information about relationships between classes. This omission limits the model's ability to generalize, particularly in zero-shot scenarios. (2) LMCL's reliance on cosine distance for embedding separation results in radiating, elongated clusters in the feature space, which are poorly suited for density-based outlier detection methods. The lack of compactness in these embeddings makes distinguishing known intents from unknown ones more challenging. To address these limitations, Fan et al. propose the Semantic-Enhanced Gaussian Mixture Model (SEG). By modeling intent embeddings with a Gaussian Mixture Model (GMM), SEG ensures that embeddings form dense, ball-like clusters, making density-based outlier detection methods like LOF more effective. Additionally, by injecting class semantic information into the GMM, SEG learns more class-concentrated embeddings, improving both intent classification and unknown intent detection.

Zhang et al. [ZXL21] improve upon previous approaches by moving away from the GMM-based method, arguing that it requires architectural modifications while failing to define explicit decision boundaries due to its reliance on LOF. Instead, they propose an Adaptive Decision Boundary (ADB) method. Using BERT to extract utterance features, they learn spherical decision boundaries for each intent class, ensuring that known intents are enclosed within their respective boundaries while unknown intents remain outside. The learned decision boundaries dynamically adjust by balancing empirical risk (ensuring correct classification of known intents) and open space risk (preventing misclassification of unknown intents), without requiring negative samples or model modifications.

A comprehensive comparison table of the described approaches can be viewed in Table A.1

2.2.2 *Intent Discovery*

While intent detection is typically framed as a classification problem, intent discovery is better understood as a clustering task. In intent discovery, similar utterances are grouped together to form intent-clusters. This can be approached in both unsupervised and semi-supervised settings. In the unsupervised setting, no prior information is available, while in the semi-supervised setting, partially labeled data is used to guide the clustering process. A comprehensive table comparing the presented approaches can be viewed in Table A.2.

2.2.2.1 *Unsupervised Setting*

Traditional clustering algorithms, such as K-Means [Mac+67] and agglomerative clustering [GK78], struggle when applied to high-dimensional data due to their inherent limitations. To address this, researchers have optimized the feature space in advance to learn compressed representations, which can improve the clustering process [Xu+15].

With the advent of deep learning, researchers began applying DNNs to the problem, enabling the simultaneous learning of feature representations and cluster assignments. This approach has shown significant improvements in performance compared to the sequential methods previously used [XGF16], [Yan+17].

In the field of CV, Contrastive Clustering (CC) has been successfully used to enhance cluster separation. This method creates positive and negative pairs through various data augmentation techniques and employs a dual contrastive learning framework at both the instance and cluster levels. The goal is to maximize the similarity between positive pairs while minimizing the similarity between negative pairs, leading to well-separated clusters [Li+21].

Building on this, Zhang et al. [Zha+21a] applied the contrastive clustering approach to short text, proposing Supporting Clustering with Contrastive Learning (SCCL). They explored several data augmentation methods under different settings: (1) *WordNet Augmenter*, which replaces words in a text with their synonyms from WordNet; (2) *Contextual Augmenter*, which uses pretrained transformers to insert or substitute suitable words; and (3) *Paraphrasing via back translation*, where a text is translated into a different language (e.g., French) and then back to English.

2.2.2.2 *Semi-Supervised Setting*

For the semi-supervised setting, Lin et al. [LXZ19] propose Constrained Deep Adaptive Clustering with Cluster Refinement (CDAC+). They criticize existing methods for relying on intensive feature engineering, which can lead to overfitting and sensitivity to the number of clusters. Instead, CDAC+ frames clustering as a pairwise classification task, determining whether two utterances belong to the same intent. It leverages BERT

to generate sentence embeddings and computes cosine similarity between intent representations. A small set of labeled intent pairs is used to define pairwise constraints. For unlabeled data, similarity labels are dynamically assigned based on thresholding: High similarities are treated as the same intent, while low similarities are treated as a different intent. Intermediate similarity pairs are left unassigned to reduce noise. This iterative process refines clustering assignments over time. Once initial clustering is complete, CDAC+ further refines cluster assignments using self-training, encouraging the model to learn from high-confidence predictions, thereby improving both representation quality and clustering stability.

Improving upon their previous work, Zhang et al. [Zha+21b] propose **Deep Aligned Clustering**. They identify two key shortcomings of CDAC+: (1) The reliance on pairwise similarities as weak supervision signals, which becomes ineffective when unlabeled data contains a mixture of known and unknown intents. (2) The difficulty in effectively transferring knowledge from known intents to new ones, leading to performance degradation as the number of new intents increases. To address these challenges, Deep Aligned enhances both knowledge transfer and clustering quality. First, the model is **pre-trained** using the limited labeled data with a softmax classification loss, ensuring well-initialized intent representations. Then, clustering is performed on the extracted intent features, and the number of clusters K is estimated by removing low-confidence clusters. To stabilize the clustering process, they introduce an **alignment strategy**. In each training epoch, k-means clustering is applied to generate cluster assignments, which are then used as pseudo-labels to train the deep neural network. However, since cluster assignments can vary across training epochs, they employ **centroid alignment** using the Hungarian algorithm to ensure consistency between successive iterations. This prevents label permutation issues and allows the model to retain historical learning information, leading to more robust clustering performance.

In 2022, Zhang et al. [Zha+22a] propose a novel approach for new intent discovery by integrating **multi-task pre-training** and **contrastive learning**. They leverage both **publicly available, high-quality intent detection datasets** and the **labeled and unlabeled utterances** from the target domain to pre-train a language model, enabling it to learn **task-specific utterance representations** for the discovery task. To improve clustering performance, they adopt a **contrastive loss function**, a technique inspired by its success in both CV and NLP. However, instead of using a standard contrastive loss, they introduce a **neighborhood-aware contrastive learning objective**, which incorporates semantic relationships between utterances. This method encourages utterances with similar intents to be grouped together while ensuring better cluster separation. Their approach significantly outperforms previous methods by **enhancing knowledge transfer**, **stabilizing clustering assignments**, and **improving representation quality** for both labeled and unlabeled intents. Experimental results on multiple intent recognition benchmarks demonstrate substantial improvements in both unsupervised and semi-supervised new intent discovery.

Despite these advancements, prior methods continued to face three key challenges: (1) Heavy reliance on labeled data, leading to severe performance drops in fully unsupervised settings. (2) Limited knowledge transfer in semi-supervised scenarios, where

leveraging a small set of labeled data remains inefficient. (3) Difficulty in estimating the number of new intent clusters, a crucial factor for real-world applicability. To address these issues, Zhang et al. [Zha+24] proposed Unsupervised and Semi-Supervised New Intent Discovery (USNID), a unified clustering framework for unsupervised and semi-supervised new intent discovery. USNID first applies unsupervised contrastive learning to pre-train on unlabeled data, constructing positive pairs through data augmentation techniques to extract high-level intent representations. This pre-training step ensures that even without labeled data, the model can learn meaningful intent structures. A major innovation of USNID is its centroid initialization strategy, which significantly improves clustering consistency. In traditional partition-based methods like k-means, cluster assignments often fluctuate across iterations, leading to instability. USNID leverages centroids from previous clustering iterations to initialize the next round, ensuring smoother convergence and more reliable cluster assignments. This alignment mechanism enhances both the efficiency and accuracy of clustering. Empirical results demonstrate that USNID outperforms previous state-of-the-art methods by 10-30% in clustering metrics such as Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). By integrating contrastive learning, centroid alignment, and self-supervised clustering, USNID establishes a new benchmark for new intent discovery, particularly in low-resource and real-world settings.

2.2.3 *Exploration of the Intent Benchmark Datasets*

To get a better overview of the data basis, we present intent benchmark datasets, commonly used for evaluation of intent detection and intent discovery models.

SNIPS. The SNIPS [Cou+18] dataset is a popular benchmark dataset used for intent classification and slot filling tasks. It was released by Snips, a voice platform company. It contains a total of 13,784 utterances across 7 intent categories. Its intents include:

- `PlayMusic` (e.g., *"play rap album one by gene vincent"*)
- `GetWeather` (e.g., *"what is the weather in sint maarten"*)
- `AddToPlaylist` (e.g., *"add rosemary clooney to pura vida playlist"*)
- `BookRestaurant` (e.g., *"i want to book the hat for my grandfather and i in arkansas"*)

Besides that, the SNIPS dataset contains annotations for *Slot Filling*, that help a model extract relevant entries. For example for an utterance such as *"Book a table for two at Joe's Diner tomorrow at 7 PM"*, the corresponding intent would be `"BookRestaurant"`, alongside different slot labels such as `"restaurant_name": "Joe's Diner"`, `"number_of_people": "two"`, `"datetime": "tomorrow at 7 PM"`. The intents of the SNIPS dataset are equally distributed, making it a balanced dataset. Finally, as this work is focusing specifically on fine-grained intents, the slot filling aspect is mostly disregarded. The SNIPS dataset is considered easy for the task of intent detection, as the categories are mostly well separated and only 7 intents in total need to be correctly distinguished.

BANKING77. The banking77 [Cas+20] dataset on the other hand, as its name suggests, belongs to the banking domain containing queries for banking and finance-related queries. It contains 13,083 customer service queries classified into 77 different intent categories. Its intents are very fine-grained including:

- Card-related issues (e.g., *"How do I activate my new card?"*)
- Transaction inquiries (e.g., *"Why was my transaction declined?"*)
- Account-related questions (e.g., *"How do I change my account password"?*)
- Security concerns (e.g., *"How can I report fraud?"*)

The dataset was manually labeled, ensuring clear and distinct intent labels with minimal ambiguity. Its utterances stem from actual user interactions from customer support logs or virtual assistant interactions. To derive a set of high-quality queries, that are clear in terms of expression, and their underlying intent, noisy and irrelevant data was removed. With respect to the annotation process, the labeling has occurred manually, by annotators familiar with banking teminology. Furthermore, efforts were made to ensure a high inter-annotator agreement, meaning that only data has been added to the dataset if multiple annotators had assigned the same label to a given data point.

CLINC150. The CLINC150 dataset [Lar+19] is a widely used benchmark for evaluating both intent classification and out-of-scope, also known as Out-of-Domain (OOD), detection in task-oriented dialogue systems. It consists of 150 fine-grained intent classes organized into 10 broad domains, such as *Banking*, *Credit Cards*, *Travel*, *Utilities*, and *Weather*.

The dataset is balanced, with each intent class containing exactly 100 examples, totaling 15,000 in-scope utterances. In addition, it includes 1,200 out-of-scope examples—utterances that do not correspond to any of the defined intent classes. This design enables robust evaluation of a model's ability to both classify known intents and correctly reject unfamiliar ones.

Representative examples from the dataset include:

- `change_pin` (e.g., *"How do I change my PIN?"*)
- `weather_query` (e.g., *"What's the weather like today?"*)
- `insurance_change` (e.g., *"I need to switch to a new insurance plan"*)
- `OOS` (e.g., *"Tell me a joke"*)

A key distinction of CLINC150 compared to other intent detection datasets is its inclusion of OOD examples. This makes it especially valuable for evaluating whether models can avoid misclassifying unseen or irrelevant queries as valid intents—an important consideration for real-world deployment of conversational agents.

STACKOVERFLOW. The StackOverflow dataset is a widely used benchmark, originally published on Kaggle.com [1]. It comprises a total of 3,370,528 samples collected between July 31, 2012 and August 14, 2012. Each sample corresponds to a question related to a programming topic, with the label indicating the associated programming language or technology.

Unlike typical intent detection tasks where the goal is to infer a user's underlying intent from an utterance, this dataset focuses on categorizing technical questions into their appropriate programming language or framework. Example categories include:

- Scala (e.g., *"Scala Regex Multiple Block Capturing"*)
- Oracle (e.g., *"Use Oracle 6 from ASP.NET application"*)
- Hibernate (e.g., *"HQL 1 to many count() question"*)
- Haskell (e.g., *"How do I test if a floating point number is an integer in Haskell?"*)

The benchmark version of the dataset contains 20 high-frequency categories that have been manually verified to ensure class distinctiveness. Each category includes exactly 100 examples, resulting in a balanced and uniform distribution across classes. Preprocessing was applied to retain only short text snippets by stripping away HTML, code, and other metadata, allowing models to focus solely on the natural language content.

While the labels reflect broad technical topics rather than fine-grained user intents, the dataset is frequently used in intent detection and intent discovery research due to the structural similarity of the task.

2.2.4 *Intent Classification in Email Data*

The previous section describes the intent detection and discovery on short text utterances. However, since the data source of this work contains the texts of emails, this sections purpose is to provide an overview for research concerning emails as a whole.

One of the earliest works in this area is by Cohen et al. [CCM04]. They analyzed different email corpora, including their own inboxes, in search for regularities to derive a coarse-grained taxonomy to classify emails into speech acts as seen in Figure 2.1. They assume that a single email may contain multiple acts (intents), with each intent being describable by a verb-noun pair drawn from the suggested taxonomy (e.g. "Deliver Data", "Propose Meeting").

Building on the taxonomy Cohen et al. proposed, Sappelli et al. [Sap+16] further refined it and used it annotate two email datasets, Enron and Avocado. They found about half of the corporate email messages to contain at least one task, mostly informational or procedural in nature. They propose their taxonomy to describe the main purpose conveyed by an email as the intent of a message, however, within this message, the sender can describe (either implicitly or explicitly) one or more tasks to be undertaken

1 https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/download/train.zip

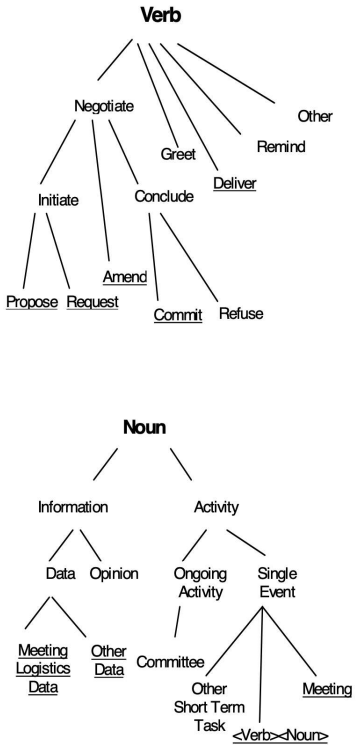

Figure 2.1: Email intent taxonomy as proposed by Cohen et al.

by the receiver. For a subset of mails, they annotated the mails in terms of the two main email acts in the message, where they adopted the verbs from the taxonomy of Cohen et al. Another dimension of their proposed taxonomy is the implicit reason of the message, which can assume one of the following categorical values: administrative procedure, legal procedure, internal collaboration, external collaboration, travel planning, employment arrangements, logistic arrangements, personal and other. For the annotated data, they found that most messages contained a single task (55.6%), with 35.6% containing no task whatsoever. The main email act for most emails was to deliver information (52.2%), followed up by requests (21.7%). The implicit reason for a sent email was mostly collaborations (43.2% internal and 34.1% external). Most tasks explicitly described in an email were informational in nature (63.3%), while most remaining tasks were procedural (30%).

Wang et al. [Wan+19] also analyzed the Avocado dataset in terms of its intents. However, instead of categorizing the email into a broad set of categories, they focus on intent-identification on sentence-level and analyzed how incorporating more context (such as the full email body and other metadata) helps improving the identification performance. For the intent taxonomy they define four overarching intent categories: (1) **Information Exchange** where the sender intends to either request or to provide information. Common uses are for example the asking of questions, requesting or sharing content, status updates, etc. They subdivide this category into *share information* and *request information*, differentiating between the provision of information vs. the requesting of information. (2) **Task Management** is also subdivided into two categories where the sender is either *requesting* an action, or is *promising* an action. (3) **Scheduling**

and Planning is subdivided into *schedule meeting* and *reminder*. Scheduling a meeting is refering to the intention of organizing a meeting (i.e., a physical meeting, a conference call or a regular phone call). Reminder on the other hand is refering to the intent of the sender reminding their recipient(s) about an upcoming meeting or event. (4) **Social Communication** are casual messages between work contacts, as well as between friends and family. As sub-intents they use *greeting messages* and *thank you notes*. Another aspect they shed light on is multiple intents vs. single intents. Emails usually contain more than one intent and the intents are not mutually exclusive, for example an email could contain a reminder about a deadline while also requesting a specific task to be completed before mentioned deadline. They find that approximately 55.2% contain a single intent, 35.8% contain two intents and 9.0% contain three or more intents. They also found that some intents such as *share information* and *request information* are more likely to co-occur.

In contrast to these prior approaches focusing on email-level speech acts, this thesis targets **fine-grained intents**, as defined in Section 1.1. Our goal is to capture more specific distinctions necessary for detailed analysis and downstream applications. Analyzing communication at this fine-grained level within the complex, conversational context of corporate email remains a relatively unexplored area.

2.2.5 *Large Language Models for Data Annotation*

With the advent of LLMs, new research areas have emerged that leverage these tools to automate tasks. One of these tasks is data annotation. This process is not trivial, as it requires domain expertise and is resource-intensive, particularly when the labeling needs to occur manually. Advanced LLMs such as OpenAI's GPT-4 [Ope+24], Meta's LLaMA [Gra+24] or Google's Gemini [Tea+24] offer a promising opportunity to revolutionize data annotation. Tan et al. [Tan+24] conducted an extensive literature review covering different annotation aspects, such as the annotation generation itself, how annotations should be assessed, as well as how they should be utilized. This section presents a mixture of the work of Tan et al., as well as research of our own. While some researchers argue that LLMs are on par or even better than human annotators [GAK23], [He+24], others highlight the shortcomings of LLMs compared to human annotators [YCS24], [Tse+24].

2.2.5.1 *Generating Labels by LLMs*

For labeling, many researchers leverage LLMs to help automate the labeling process. Chen et al. [Che+24] employ LLMs as expert annotators for event extraction. Whereas most research in this area tries to replace human annotators by LLMs, Li et al. [Li+23] propose a collaborative paradigm called *CoAnnotating*, where LLMs are being used to assist humans with the labeling task, rather than to replace them. Choi et al. [Cho+24] use multiple LLMs to generate different outputs on the same task, imitating how labeling tasks would be conducted using human annotators, by generating multiple labels and letting the majority vote decide what the label should be.

2.2.5.2 *Assessing LLM-Generated Annotations*

A general approach to evaluate LLM-generated labels is to compare them against labels created by human annotators. Regarding the adherence of the LLM following a set of given guidelines, Efrat et al. proposed the "Turking Test" [EL20]. In scenarios where extensive datasets are required, the quality of LLM-generated annotations is compared to a small subset of the dataset, that is manually labeled, making up the gold standard [Zha+22b], [Agr+22], [He+24]. Wang et al. [Wan+24b] propose a collaborative approach using an LLM to first annotate data, followed by a human to assess the labels to subsequently reannotate if a verification score is too low. Wan et al. [Wan+24a] calculate the accuracy of the LLM by using a pairwise comparison task, using an utterance and its corresponding label as predicted by the LLM, alongside of another label, the LLM is being prompted again to decide which label is more fitting.

2.2.5.3 *Utilizing LLM-Generated Annotations*

The annotations generated by the LLM can be utilized further for different downstream tasks. One of these is Supervised Fine-Tuning. Huang et al. [Hua+23] propose an approach of self-evolution where the LLM is used both as a data annotator, as well as a learnable model. Using the self-annotated data, the model is iteratively fine-tuned. For efficiency reasons, many studies aim to use data generated by larger models to train smaller models on them.

In-Context Learning (ICL) is another area where LLM-generated annotations are being leveraged. ICL consists of three components: a prompt (a task description for the LLM), several in-context samples (or demonstrations), as well as the test case the LLM needs to infer. Current studies leveraged LLM-generated annotations to refine or augment all of these components. Zhou et al. [Zho+22] first showed that LLMs can be used to design accurate task-descriptions rivaling the expertise of human-level prompt engineers. Demonstration augmentation has also been proven useful, especially in situations where labeled data is scarce, where provided demonstrations are being enriched and diversified using an LLM [Kim+22]. Regarding the test sample that needs to be inferred, possible augmentations are to use the LLM to rephrase the test sample once or multiple times, or to polish the orignal sample or to decompose it into several sub-questions.

3

METHODOLOGY

This chapter outlines the comprehensive methodology developed to discover, categorize, and label fine-grained user intents within the complex domain of corporate email, specifically using the Avocado Research Email Collection. Addressing the inherent challenges of analyzing large-scale, real-world communication data—where intents are often implicit, overlapping, embedded in noisy text, and lack a predefined taxonomy—required moving beyond standard classification or simple clustering techniques applied directly to raw text. The approach detailed herein evolved through exploration and iteration, aiming for a scalable and systematic process to navigate these complexities.

We begin by introducing the Avocado dataset, the foundation of this study, and detail the crucial preprocessing steps undertaken to handle its scale, varied formats, and inherent noise (Section 3.1). This section also recounts initial exploratory data analyses, including action-object pair extraction and early clustering attempts on raw text, highlighting the limitations encountered (such as topic mixing and sensitivity to non-intent features like signatures) that necessitated a more sophisticated strategy.

Subsequently, Section 3.2 presents the core contribution of this chapter: the proposed multi-stage workflow designed specifically for fine-grained intent discovery and labeling in this domain. **A foundational decision shaping this workflow is the focus on identifying intents primarily at the sentence level.** This approach was chosen deliberately to manage the complexity arising from emails often containing multiple sentences that serve distinct communicative functions or express several intents simultaneously. Analyzing entire emails for a single primary intent would obscure these nuances and grapple directly with the difficult problem of multi-intent representation from the start. By concentrating on individual sentences, particularly those filtered for clarity and self-containment (Section 3.2.1), we aim to isolate more discrete intent expressions, making the discovery and labeling task more tractable and aligning with the utterance-level format common in many intent classification benchmarks. This focus allows for a detailed exploration of fine-grained distinctions, although it intentionally simplifies the problem by initially disregarding the influence of broader email context. Operating on this sentence-level foundation, the workflow integrates several key techniques:

- Targeted data filtering using a combination of linguistic rules and LLM-based quality scoring to isolate high-potential, self-contained intent-bearing sentences (Section 3.2.1).
- Leveraging a LLM (LLaMA 3 8B Instruct) not merely for labeling, but for generating richer, intent-focused semantic features—including explicit intents, implicit intents, and purpose summaries—for each utterance (Section 3.2.2).

- Employing semantic clustering on these generated features, rather than raw text embeddings, to group utterances based on their underlying communicative goals (Section 3.2.2).
- An iterative refinement and dataset expansion strategy, alternating between populating known intent categories using supervised methods and similarity search, and re-clustering the remaining data to discover potential new intents, all guided by human-in-the-loop validation (Section 3.2.3).

Finally, recognizing the importance of evaluating the outcomes of this process, Section 3.3 defines the evaluation strategy employed throughout the thesis. It outlines the specific metrics and rationale for characterizing the structural properties of the generated labeled dataset, assessing the quality and consistency of its labels (addressing RQ3), and establishing a basis for comparison against standard benchmarks and algorithm performance, setting the stage for the experimental setup (Chapter 4) and results (Chapter 5) chapters that follow. The overall methodology aims to provide a structured, semi-automated framework adaptable for tackling fine-grained intent discovery in similar complex, real-world communication datasets.

3.1 PRE-PROCESSING THE DATASET AND EXPLORATIVE DATA ANALYSIS

The foundation for the analyses presented in this thesis is the *Avocado Research Email Collection* [Dou+15], in the following referred to as 'Avocado'. Avocado was an Information Technology software and services firm developing products for the mobile Internet market, operating from the late 1990s to the middle of the first decade of the 21st century. The dataset originates from Personal Storage Table (PST) files linked to 282 accounts. However, three of these accounts yielded no data, resulting in a final collection of 279 accounts. Each account is referred to as a "custodian," which can represent both individual users and non-human entities. The collection is organized into two main components: metadata and text.

- **Metadata**: The metadata is stored in XML format and includes a top-level XML file listing all custodians, along with individual XML files for each custodian. These files detail items extracted from the corresponding PST files.
- **Text**: The text component contains the extracted content from each custodian's folders, with each item's text stored in separate files. These files are compressed into ZIP archives, grouped by custodian.

Datasets derived from real-world scenarios are immensely valuable, as downstream applications are typically designed to reflect real-world conditions. However, due to privacy concerns and regulations, publicly available email datasets are relatively scarce. The most prominent publicly accessible corporate email dataset is the *Enron Email Dataset* [KY04], which was made available as part of an internal investigation following the corporate collapse of Enron.

The handling of the Avocado dataset presented several challenges. Initially, a top-level XML file was parsed, as it contains entries for all custodians and their corresponding ZIP files, which house the actual data. For easier processing, these files were saved locally as Parquet dataframes.

Regarding the metadata, this thesis primarily focuses on the text content of the emails, rather than incorporating additional metadata into the analysis. The varied formats of the emails presented significant challenges during processing. Often, the text content was embedded within metadata, which had to be extracted to obtain the relevant information. An email parser[1] was highly beneficial in facilitating the extraction of the email body text. However, creating regular expressions to filter out specific patterns, such as email signatures and other extraneous metadata, was a time-consuming but essential task.

Additionally, the use of BERTopic [Gro22] proved valuable for identifying automatic status messages, spam emails, and advertisements—elements that were irrelevant to the focus of this thesis, which centers on human communication. Following the pre-processing steps, the dataset was reduced from the original 938,035 emails to 488,314 emails for further analysis.

In order to conduct preliminary exploratory data analysis, each email in the dataset was segmented into individual sentences using the spaCy library[2]. The resultant histogram, which visualizes the distribution of sentence frequencies, is shown in Figure 3.1. The majority of emails contain between 1 to 5 sentences. However, there are notable exceptions, particularly in emails containing more than 10 sentences. Upon closer examination, these outliers can frequently be attributed to anomalies in spaCy's sentence segmentation process. Specifically, the presence of numerous asterisks in some emails led spaCy to erroneously interpret each asterisk as a sentence boundary, thereby artificially inflating the sentence counts. Furthermore, some emails resembled chat logs, consisting primarily of disjointed utterances, which further contributed to the unusual distribution patterns observed.

To inspect the dataset and get a feeling of what kind of intents it encompasses, we leverage spaCy for dependency parsing to extract Action-Object (also known as Verb-Object) pairs from the emails. Subsequently we sorted the pairs based on their occurrence frequency in descending order. As a next step we aggregated them based on specific words, as well as words carrying only little descriptive information.

The intents we expect to find based on just the action-object pairs we inspect in the tables are the expected communications in a corporate environment. These include requests for contact such as calls and mails.

Assistance and offers or requests for help also seem to be included in the dataset.

Other than that the only other action-object pair that alludes to a specific intent topic is meeting related, potentially implying the proposal or a request for a meeting.

Other identified pairs suggest no intents whatsoever and rather imply social phrases.

To further complement the insights from action-object pairs and gain a broader perspective on potential structures within the data, preliminary clustering experiments

1 https://github.com/alfonsrv/mail-parser-reply
2 https://spacy.io/

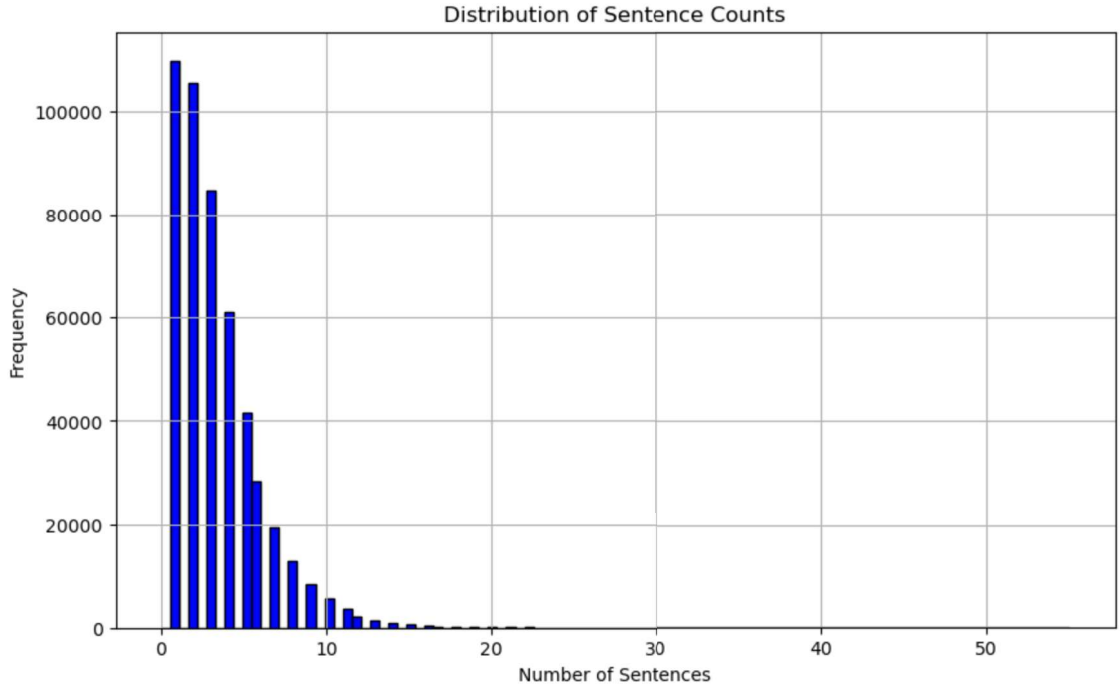

Figure 3.1: The distribution of sentence frequency in the analyzed emails.

were performed. These experiments applied agglomerative clustering to the semantic embeddings of a subset of sentences previously filtered to contain request-implying expressions (e.g., "can you," "please," "could you"). The goal was to explore whether natural groupings corresponding to intents would emerge based purely on semantic similarity within this targeted subset.

Initial inspection of the resulting clusters revealed groups centered around common corporate themes, some of which superficially resembled broad intent categories, such as meeting arrangements or document requests. However, closer examination of the utterances *within* these seemingly coherent clusters often exposed a significant mixing of distinct communicative goals. For example, a cluster broadly related to 'meetings' might contain sentences explicitly *requesting* a meeting, alongside sentences asking for information *about* an existing meeting (e.g., the time or location), or even confirming attendance. While topically related, these represent different fine-grained intents.

This finding underscores a key limitation of applying standard semantic clustering directly for fine-grained intent discovery in this domain: such methods tend to group utterances based on broader **topical similarity** rather than the specific underlying **communicative purpose** or desired action. Consequently, these preliminary results highlighted the necessity for a more specialized approach, one capable of generating features that emphasize the user's specific goal, to achieve meaningful clustering and ultimately enable the discovery and labeling of distinct, fine-grained intents. This motivates the LLM-based feature generation and iterative refinement workflow detailed in the subsequent sections.

3.2 PROPOSED INTENT DISCOVERY AND LABELING WORKFLOW

The proposed approach involves three main processing steps that are being applied to the text data: (1) a filtering step involving rules, as well as scores assigned by LLaMA, to obtain a set of well articulated utterances, clearly expressing their intent, referenced in Section 3.2.1. (2) an automated annotation step, leveraging LLaMA, to obtain important features for the utterances, mandatory to yield intent clusters, referenced in Section 3.2.2. (3) the manual inspection of the resulting intent clusters for inclusion in our intent dataset, followed by a subsequent step to further populate existing classes and to iteratively refine the intent taxonomy, referenced in Section 3.2.3.

3.2.1 *Filtering for quality utterances*

In order to facilitate the intent discovery process, we decided to focus on single label intents on a sentence-level instead of analyzing emails as a whole. By this we deliberately disregard context from previous messages or other sentences surrounding the target sentence for which we want to identify the intent. As intent benchmark datasets usually contain single sentence utterances with one specific request, we aim to extract comparable utterances from the avocado dataset.

RULE-BASED FILTERING. To focus on sentence-level intents, the first step is to split each email into its individual sentences. This is accomplished using spaCy, which divides the 279,819 emails in the training set into 923,929 individual sentences. Next, a rule-based filtering approach is applied. Since most request-type sentences fall within a medium sentence length range, we retain sentences with a word count between 5 and 15 words, inclusive. This filter removes short phrases like "thank you" while ensuring that sentence length remains concise by excluding sentences with more than 15 words. This further reduces the total number of sentences to 500,788.

To refine the focus on sentences that likely contain an intent, we apply textual patterns indicative of specific requests, such as "please," "can you," and "could you", the exhaustive list of the used expressions implying a request is to be found in Table A.3. Sentences that contain at least one of these patterns are treated as candidates containing a request and are retained for further analysis. The pattern list is expanded using ChatGPT to identify similar phrases, resulting in a total of 34 patterns. This further narrows down the pool of sentences likely expressing an intent to 49,759.

SCORE-BASED FILTERING. As the utterances from the previous step are merely candidates for containing intents, we utilize `LLaMA 3 8B Instruct` [3] to apply score-based filtering as the next step in our processing pipeline. For this, LLaMA is being prompted to assign different scores, ranging from 1-5, with 5 being the best score, to each utterance, based on how effectively an intent is expressed in each sentence. The scoring criteria are as follows: *Intent Clarity*, which evaluates how explicitly and effectively the

3 https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

intent is articulated, favoring sentences that are free from ambiguity. *Self-Containment*, which ensures that a sentence is independent and does not rely on external context, as we aim to extract utterances comparable to the data that is being used in traditional intent detection benchmarks. The final scoring criterion is *Specificity in Task and Object Definition*, which assesses whether entities, tasks, and objects are clearly defined rather than referenced with vague terms such as "them" or "it".

After obtaining the quality scores by LLaMA, we only keep sentences that have been assigned the maximum score for each criterion. By this we arrive at a final set of 19,450 well-articulated utterances containing a clear intent. The filtering approach is visually expressed in Figure 3.2

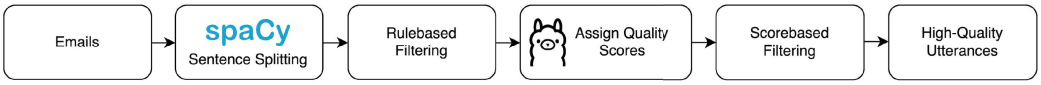

Figure 3.2: The emails are first split into individual sentences using spaCy. The resulting set of sentences are subsequently filtered using rules to retain sentences with a high likelihood of containing an intent. For the penultimate step LLaMa is being leveraged to assign different scores, for which a final filtering is being applied to derive high quality utterances.

The resulting prompt from the described approach is detailed in Listing C.1. The sentences are being injected dynamically into a variable within the prompt template where the LLM is analyzing each sentence.

3.2.2 *LLM-based Annotation and Clustering*

To analyze communicative functions within the Avocado dataset at scale, a method for identifying user intents was required. Manual annotation of the entire dataset was deemed infeasible due to its size. Furthermore, preliminary analysis suggested that utterances expressing similar underlying intents often lack direct surface-level semantic similarity, making simple clustering of raw utterances ineffective. Therefore, an automated approach was developed leveraging LLaMA to generate intent-focused textual features, which could then be clustered based on semantic meaning, visualized in Figure 3.3. This approach evolved through several iterations, detailed below, to address challenges encountered in achieving coherent and meaningful intent groupings.

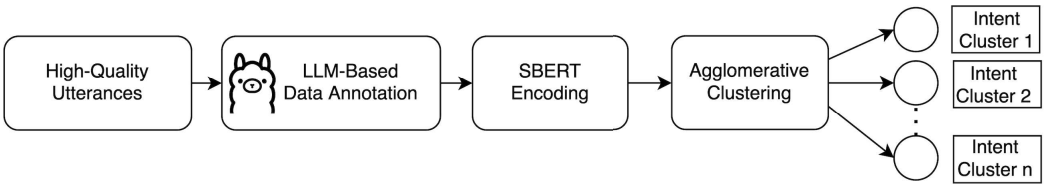

Figure 3.3: Process for generating intent clusters: High-quality utterances (filtered according to Figure 3.2) are annotated using LLaMA. Subsequently, SBERT generates semantic embeddings for these annotations, which are then clustered to produce fine-grained intent clusters.

FIRST ITERATION: SINGLE INTENT DESCRIPTORS. The initial strategy involved prompting an LLM to analyze each relevant utterance and generate a single, concise "intent descriptor", summarizing the sender's primary goal (e.g., `schedule_meeting`, `request_contact_method`). The hypothesis was that these descriptors would abstract away surface variations while preserving core intent, allowing utterances with similar goals to be grouped via semantic clustering of the descriptor embedding.

However, when these generated descriptors were conceptually embedded and clustered based on semantic similarity, the resulting clusters revealed significant limitations. While some groupings were sensible (e.g., various ways of scheduling a meeting clustered together), many were heterogeneous and failed to capture distinct intents accurately. Key problems included:

- **Descriptor Ambiguity**: Descriptors like `request_copy` were too broad, leading to the erroneous merging of distinct actions such as requesting a document copy with requesting to be copied on an email. Similarly, `request_direction` conflated requests for task instructions with requests for navigational directions.
- **Implicit Intent Neglect**: The single descriptor often captured only the explicit statement, missing underlying implicit intents.

These results indicated that single, short descriptors lacked the necessary specificity to reliably represent user intent for fine-grained clustering.

SECOND ITERATION: INCORPORATING IMPLICIT INTENT DESCRIPTORS. To specifically address the problem of overlooked implicit meanings, the prompting strategy was refined. The LLM was instructed to generate *two* distinct features for each utterance: one descriptor capturing the *explicit* intent and another capturing the *implicit* intent (if present). The rationale was that combining representations of both facets might lead to better differentiation during clustering.

While this approach succeeded in capturing implicit intents noted as missing from the previous iteration, preliminary evaluations suggested that clustering based on these two potentially short descriptors still suffered from the ambiguity issues observed previously. The fundamental problem of descriptors lacking sufficient context remained partially unresolved.

FINAL ITERATION: INCLUDING PURPOSE SUMMARIZATION (FINAL APPROACH). Recognizing that concise descriptors struggled to capture the full nuance and context of an utterance, a third, more descriptive feature was introduced. The LLM prompt was further enhanced to generate a brief, sentence-level 'purpose summarization' of the utterance, explicitly focusing on the user's overall goal or the disired action resulting from the message.

Conceptually clustering based on embeddings derived *solely* from the purpose summaries yielded markedly more coherent and interpretable intent groups than the previous iterations. The richer context provided by the summaries appeared crucial for disambiguation.

To leverage the strengths of all generated features, the final adopted methodology involved combining representations from all three: the explicit intent descriptors, the implicit intent descriptors, and the purpose summarization. Embeddings were conceptually generated for each, and these were concatenated to create a composite feature vector for each utterance. This multi-feature representation formed the basis for the final clustering process. This approach was found to produce the most consistent, fine-grained, and semantically meaningful intent clusters, overcoming the major limitations identified in earlier iterations. The corresponding prompt is referenced in Listing C.2.

3.2.3 *Iterative Dataset Expansion and Refinement*

The initial intent clusters derived from the LLM-based annotation and clustering process provided a foundational seed set of labeled utterances for identified intents. To systematically build upon this foundation and construct a comprehensive labeled dataset, we employed an alternating, iterative strategy (illustrated in Figure 3.4). Each iteration involved two main phases: (1) expanding the labeled examples for known intents, and (2) exploring the remaining unlabeled data for potential new intents.

PHASE 1: EXPANDING KNOWN INTENTS The core automated techniques used to identify candidate utterances for manual review included:

- **Semantic Classification (Supervised Approach)**: This involved training classification models on the *currently available labeled data* (starting with the seed set and growing with each iteration) to predict labels for unlabeled utterances. High-confidence predictions were flagged as strong candidates belonging to existing classes. Two main modeling approaches were utilized:
  - **SetFit**: This approach [Tun+22] leverages pre-trained sentence transformers (like SBERT) for embedding generation. It first employs contrastive learning (using labeled examples) to fine-tune the transformer body, encouraging better separation of known intent classes in the embedding space. Subsequently, a lightweight classification head (e.g., Logistic Regression) is trained on these fine-tuned embeddings. The key advantage is its effectiveness even with limited labeled data per class. When applied to the unlabeled pool, SetFit models produce predicted labels and associated confidence scores. Utterances predicted with high confidence (score > 0.8) for a known intent were selected as candidates for that intent.
  - **Adaptive Decision Boundary (ADB)**: The ADB method [ZXL21] was adapted from its original implementation[4] for identifying high-confidence examples. ADB, using BERT-based features, learns spherical decision boundaries for each known intent class during training. When applied to unlabeled data, instances falling well within the learned boundary of a specific class (i.e., far from the decision boundary shared with the "unknown" space or other

4 Based on https://github.com/thuiar/TEXTOIR

classes) can be considered high-confidence examples of that known intent. These were also selected as candidates.

- **Cosine Similarity Search (Embedding-Based Approach)**: Sentence embeddings were generated for representative utterances already labeled within each known intent category (e.g., cluster centroids or manually selected exemplars). These embeddings were then compared against the embeddings of all unlabeled utterances using cosine similarity. Unlabeled utterances exhibiting very high cosine similarity (e.g., above a threshold of 0.9) to the exemplars of a specific intent class were flagged as strong candidates for that class.
- **Rule-Based Filtering (Pattern-Based Approach)**: Based on linguistic patterns observed in the confirmed examples for certain intents (e.g., specific keywords, phrases, or sentence structures identified during manual review), targeted regular expressions were crafted. These rules were applied to the unlabeled pool to efficiently surface utterances matching these specific structural patterns, providing another source of candidates, particularly for intents with distinct syntactic markers.

PHASE 2: DISCOVERING POTENTIAL NEW INTENTS VIA RE-CLUSTERING After expanding the known categories and thus thinning the unlabeled pool, the second phase focused on identifying potentially novel intents that might have been obscured initially. To achieve this:

- The remaining unlabeled utterances were isolated.
- The original clustering methodology (as detailed in Section 3.2.2) was re-applied specifically to this reduced, unlabeled subset.
- The resulting clusters were manually inspected to identify any new, coherent groupings that emerged and represented distinct communicative goals not covered by the existing taxonomy.
- If such new intents were validated, representative seed examples were manually selected from these clusters and added to the labeled dataset, effectively introducing new intent categories into the taxonomy.

ITERATIVE CYCLE AND REFINEMENT. This two-phase process (expand known, discover new) was repeated iteratively. In each subsequent cycle, the expansion techniques (Phase 1) benefited from the larger, more diverse labeled dataset (including any newly discovered intents from Phase 2 of the previous cycle). Similarly, the re-clustering (Phase 2) operated on a progressively smaller and potentially more distinct pool of remaining unlabeled data, increasing the chance of uncovering less frequent or more nuanced intents. By alternating between clustering unlabeled data and applying open-intent detection methods, the intent dataset can be efficiently and iteratively constructed. This process is illustrated in Figure 3.4 and described as an algorithm in Algorithm 1.

Manual verification at each stage was paramount. This alternating strategy allowed for both the deepening (more examples per intent) and broadening (discovery of new intents) of the labeled dataset efficiently, balancing automated suggestion with essential human oversight.

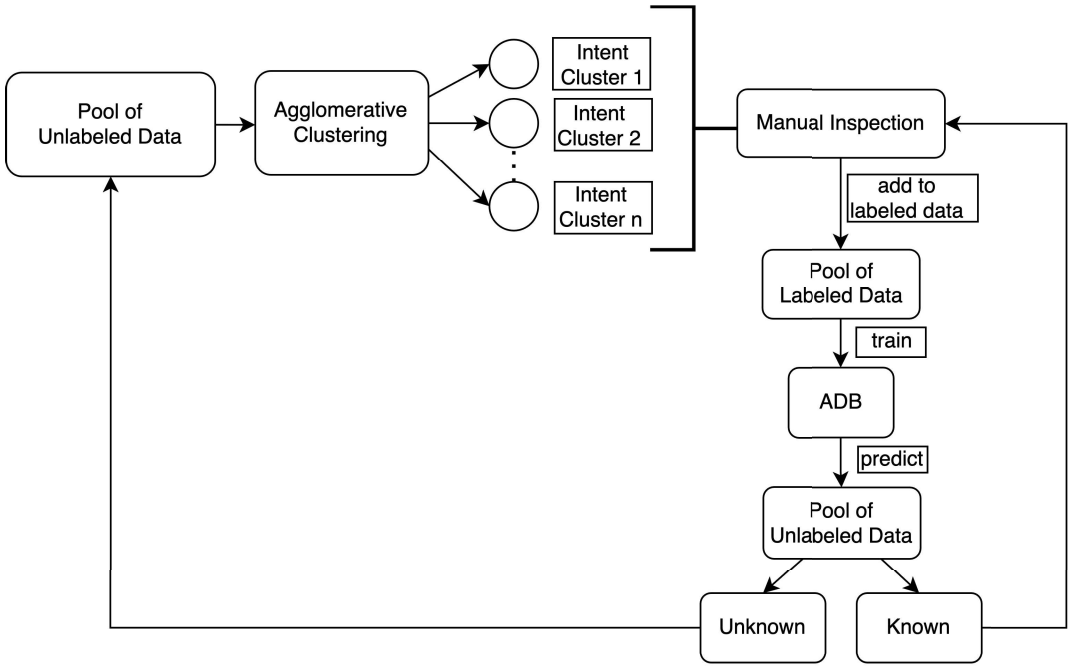

Figure 3.4: Proposed Approach: To retrieve intent clusters, the unlabeled data is first clustered based on annotations generated by the LLM, as illustrated in Figure 3.3. After manual inspection using a human-in-the-loop process, an initial set of labeled data is derived. This labeled data is then used to train the ADB Open Intent Classifier, enabling the identification of additional samples for known classes. As more data is added to the labeled set, the pool of unlabeled data is gradually reduced. Clustering is then reapplied to the remaining unlabeled data to repeat the process of discovering new intent categories, as well as adding more diverse examples to existing categories, further expanding the dataset and ensuring a diverse set of samples for each intent category.

3.3 EVALUATION STRATEGY

Following the description of the proposed methodology for intent discovery and dataset labeling (Section 3.2), this section outlines the comprehensive evaluation strategy employed in this research. Evaluating the outcome involves multiple facets: characterizing the structural properties of the generated labeled dataset, assessing the quality of its labels, contextualizing its characteristics through comparison with established benchmarks, and measuring the performance of standard algorithms upon it. The following subsections define the specific metrics and approaches used for each aspect of this evaluation, forming the basis for the procedures detailed in Chapter 4 and the results presented in Chapter 5.

Algorithm 1 Iterative workflow for intent discovery and labeling

Input: Unlabeled dataset $\mathcal{U}$
Output: Labeled dataset $\mathcal{L}$ with discovered intent categories
Initialize $\mathcal{L} \leftarrow \emptyset$ ▷ Start with an empty labeled set
$i \leftarrow 0$ ▷ Initialize iteration counter
while $|\mathcal{U}| > 0$ **do** ▷ Repeat until all data is labeled
  if $i \bmod 2 = 0$ **then** ▷ Alternate between clustering and model training
    Generate clusters from $\mathcal{U}$ using LLM annotations
  else
    Train ADB model using $\mathcal{L}$
    Use trained ADB model to predict labels for $\mathcal{U}$
  end if
  Perform human-in-the-loop verification to obtain $\mathcal{L}_{new}$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}_{new}$ ▷ Add new labels
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{L}$ ▷ Remove labeled data from the unlabeled set
  $i \leftarrow i + 1$ ▷ Increment iteration counter
end while
return $\mathcal{L}$

3.3.1 *Dataset Characterization Metrics*

INTRA-CLASS SIMILARITY. For each intent class, all pairwise cosine similarities are calculated among the embedding vectors corresponding to utterances with the same label. Let $\{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ denote the set of embeddings for utterances belonging to a class c. The intra-class similarity is defined as the average cosine similarity across all distinct pairs:

$$\text{IntraSim}(c) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \cos(x_i, x_j)$$

To avoid inflating the similarity score, self-similarity values on the diagonal of the similarity matrix are excluded. In addition to the mean, the variance of the pairwise similarities is also computed to capture the internal spread or semantic tightness of each class in the embedding space.

INTER-CLASS SIMILARITY. To estimate the semantic proximity between different intent classes, average pairwise cosine similarity is computed between all utterances from two distinct classes. For each unique class pair (c_i, c_j), let $\{x_1^{(i)}, \ldots, x_m^{(i)}\}$ and $\{x_1^{(j)}, \ldots, x_n^{(j)}\}$ represent the embedding sets for classes c_i and c_j, respectively. The inter-class similarity is then given by:

$$\text{InterSim}(c_i, c_j) = \frac{1}{mn} \sum_{k=1}^{m} \sum_{l=1}^{n} \cos\left(x_k^{(i)}, x_l^{(j)}\right)$$

This formulation results in a symmetric similarity matrix across all class pairs, allowing for the identification of semantically overlapping intents, which may pose challenges for intent classification due to their close proximity in embedding space.

3.3.2 *Internal Clustering Metrics*

SILHOUETTE SCORE. The Silhouette Score is a metric that quantifies how well a sample is clustered by comparing its intra-class similarity to its nearest inter-class similarity. Instead of solely quantifying the quality of a clustering, we leverage this metric to measure how well separated the clusters are by using the ground-truth labels of a benchmark dataset as cluster assignments. For a given utterance embedding x, let $a(x)$ denote the average distance between x and all other embeddings in the same class (i.e., intra-class dissimilarity), and $b(x)$ the minimum average distance between x and the embeddings of any other class (i.e., the closest neighboring cluster). The silhouette score $s(x)$ for a single sample is then defined as:

$$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$$

This score ranges from -1 to 1, where values close to 1 indicate that the sample is well-matched to its own cluster and clearly separated from others. A score near 0 suggests overlapping clusters, and negative values imply potential misclassification. Traditionally, the silhouette score is computed using Euclidian distance; however, we adapt it to use cosine distance instead, defined as $d_{\cos}(x, y) = 1 - \cos(x, y)$. This adaptation is motivated by the nature of high-dimensional embedding spaces, where cosine similarity is generally more meaningful than Euclidian distance due to the curse of dimensionality. Cosine-based distances better capture angular relationships between vectors, which are more indicative of semantic similarity in the context of language representations. As such, using cosine distance yields a more reliable assessment of clustering quality for utterance embeddings.

DAVIES-BOULDIN INDEX. The DBI is an internal evaluation metric used in clustering that measures the degree of separation between clusters. Although it shares similarities with the Silhouette Score, the DBI specifically quantifies the average similarity between each cluster and its nearest neighbor. This similarity is defined by the ratio of the sum of intra-cluster distances to the inter-cluster distance. For any two clusters i and j, S_i represents the average distance between all points within cluster i and its centroid, while M_{ij} denotes the distance between the centroids of clusters i and j. The DBI for cluster i is calculated as:

$$DB_i = \max_{i \neq j} \left(\frac{S_i + S_j}{M_{ij}} \right)$$

The overall DBI is then determined by averaging the DB_i values for all clusters. A lower DBI indicates superior clustering quality, suggesting that the clusters are both internally compact and well-separated from each other. Thus, the metric provides insights into how well the clusters are separated. Similar to the Silhouette Score, DBI is traditionally computed using Euclidian distances. However, just like we did for the Silhouette Score, we adapt this metric to use cosine distances instead.

3.3.3 *Label Quality Assessment using Confident Learning (Cleanlab)*

Given that the labels for the corporate email intent dataset are generated through a semi-automated workflow involving LLM annotations and clustering, rather than constituting verified ground truth, assessing their quality and identifying inconsistencies is crucial (RQ3). Standard evaluation against a gold standard test set is not directly applicable for evaluating the inherent quality of these generated labels across the entire dataset.

Therefore, this research employs **Confident Learning**, a framework designed to find label errors in datasets by analyzing the relationship between potentially noisy labels and predictions from a trained classifier. Specifically, the **Cleanlab Python library**, which implements Confident Learning algorithms, is utilized.

The core idea is to estimate the joint distribution between noisy labels and true (but unknown) labels by leveraging predicted probabilities generated from models trained via cross-validation (ensuring probabilities are out-of-sample). Cleanlab uses these estimates to identify data points where the provided label is statistically unlikely given the model's predictions, ranking them by confidence.

Cleanlab was chosen as the primary method for label quality assessment in this thesis, because it provides a **systematic and quantifiable** approach to identifying potential label noise **without requiring a separate, verified ground truth dataset**. Furthermore, it is **model-agnostic**, allowing integration with the classification models (like SetFit/Logistic Regression) already used in the workflow. Finally, it helps **diagnosing the types of inconsistencies** present, offering insights into the challenges of the labeling task and potential areas for taxonomy refinement.

The specific procedures for generating the necessary out-of-sample probabilities via cross-validation and executing the Cleanlab analysis are detailed in the Experimental Setup, Section 4.2.3.

3.3.4 *TEXTOIR evaluation*

To assess how state-of-the-art algorithms perform on the generated corporate email intent dataset and to contextualize its characteristics, evaluations were conducted using the standardized **Text Open Intent Recognition (TEXTOIR) framework** [Zha+21c]. This framework was chosen for its inclusion of established algorithms relevant to the tasks of **open intent detection** and **semi-supervised intent discovery** (as discussed in Section 2.2), facilitating reproducible and comparable experiments.

While open intent detection usually relies on metrics like accuracy and F1-scores for known and unknown classes, evaluating the quality of **intent discovery** requires metrics that compare the algorithmically generated clusters against the ground truth intent labels. For this purpose, the TEXTOIR framework utilizes, and we will report on, standard clustering metrics, primarily:

- **NMI:** This metric measures the agreement between the discovered cluster assignments and the ground truth intent labels by quantifying their shared information, normalized typically to a scale between 0 (no mutual information beyond chance) and 1 (perfect correlation). Higher NMI values indicate better clustering quality.
- **ARI:** ARI assesses the similarity between cluster assignments by considering all pairs of samples and whether they are grouped together consistently with the ground truth labels, correcting for chance agreement. ARI typically ranges from 0 (random agreement) to 1 (perfect agreement), with higher values signifying better clustering performance.

Utilizing TEXTOIR allows for evaluating algorithm performance under controlled conditions that simulate real-world challenges, such as operating with incomplete knowledge of all possible intent classes (controlled via the **Known Intent Ratio (KIR)**) or learning from limited supervision (controlled via the **Labeled Ratio (LR)**). By employing this framework, we aim to benchmark the performance of standard methods on our specific dataset using the aforementioned metrics (and others like F1-score where appropriate) and gain insights into its difficulty and data requirements for effective model training. The specific algorithms, dataset configurations, parameter settings (KIR, LR variations), and detailed reporting of these metrics used within the TEXTOIR experiments are detailed in the Experimental Setup (Section 4.2.4).

4

EXPERIMENTAL SETUP

Following the methodology for intent discovery and labeling (Chapter 3) and the evaluation strategy defined therein, this chapter describes the concrete experimental setup. It specifies the final configuration of the generated labeled dataset and the benchmark datasets used for comparison. Furthermore, it details the exact procedures followed for dataset characterization (including distribution, semantic structure, and lexical analysis), comparison against benchmarks, and label quality assessment using Cleanlab. These details provide the necessary context for the results reported in Chapter 5.

4.1 DATASETS USED FOR EVALUATION

The datasets subject to our evaluation are the labeled dataset we created ourselves, as well as several intent benchmark datasets.

4.1.1 *Labeled Corporate Email Intent Dataset*

The primary dataset analyzed in subsequent chapters was constructed using the methodology detailed in Chapter 3. It is important to note that the initial data corpus was partitioned into training, validation, and test subsets prior to the intent discovery and labeling phases. The methodology was then applied primarily within these splits. Consequently, the resulting dataset represents the outcome of this process and consists of 6,785 labeled utterances.

During the analysis and finalization of the dataset across the predefined splits, a taxonomy of 54 distinct in-scope intent classes was established based on recurring themes found primarily in the training data (full taxonomy in Table A.4).

An OOD category was utilized specifically during the preparation of the test set. If certain intent categories identified and labeled within the training split had insufficient or zero representation in the initially allocated validation or test splits, utterances corresponding to those scarce categories were re-assigned to the OOD class within the test set. This approach ensures that the evaluation reflects performance on well-represented intents while acknowledging the presence of other discovered categories that were too infrequent for robust split-wise evaluation.

The final splits used for subsequent model training and evaluation, reflecting this handling of low-frequency intents, are:

- **Training Set**: 4,804 utterances (70.8%) (Containing 54 in-scope labels)
- **Validation Set**: 1,229 utterances (18.1%) (Containing 54 in-scope labels)
- **Test Set**: 752 utterances (11.1%) (Containing 54 in-scope labels + the OOD category)

While the dataset was randomly partitioned into a 60/20/20 split, the final label distribution within these splits reflects the intents discovered and labeled via the applied methodology, including the reassignment of rare intents to OOD in the test set. Therefore, this distribution (analyzed in Section 5.1.2) **should not be interpreted as representing the true, underlying intent distribution of the original email corpus, nor is the discovered taxonomy guaranteed to be exhaustive**. To supplement the main analysis, a separate investigation was conducted in order to approximate the intent distributions found in each respective dataset partition. 100 utterances were randomly selected from each of the original unlabeled training, validation, and test partitions. These 300 samples were manually labeled by the author using the final intent taxonomy (Table A.4). These labels served only for the analysis reported in Section 5.1.2 and were not incorporated into the primary labeled dataset. A notable characteristic is a significant class imbalance within the final labeled splits, which influenced subsequent analysis steps (e.g., using `class_weight='balanced'`).

4.1.2 *Benchmark Intent Datasets*

To contextualize the characteristics of the labeled corporate email dataset developed in this thesis, its structural properties (analyzed in Chapter 5) will be compared against several established intent benchmark datasets. The following datasets, previously introduced in Section 2.2.3, were selected based on their common usage (e.g., inclusion in the TEXTOIR repository) and diverse characteristics, providing varied points of comparison.

For all benchmark datasets, the versions and standard train/validation/test splits as provided within the TEXTOIR framework were utilized directly. Key characteristics relevant to the comparison include:

- **SNIPS** [Cou+18]:
  - **Role**: Represents a dataset with a small number (7) of mostly distinct intent categories across several domains, ranging from restaurant reservations to music consumption.
  - **Size**: Contains 14,484 utterances, split approx. 90% train / 5% validation / 5% test.
- **BANKING77** [Cas+20]:
  - **Role**: Represents a fine-grained dataset within a single domain (banking), featuring 77 distinct intents.
  - **Size**: Contains 13,083 utterances, split approx. 70% train / 10% validation / 20% test.
- **CLINC150** [Lar+19]:
  - **Role**: Represents a large-scale, fine-grained dataset spanning 10 domains with 150 intents, crucially including OOD examples (though OOD examples are typically excluded from structural comparisons like this unless specified).

  - **Size (without OOD)**: Contains 22,500 in-scope utterances, split 80% train / 10% validation / 10% test.
  - **Size (OOD)**: Contains 23,700 utterances, 22,500 in-scope utterances with additional 1,200 OOD utterances, split approx. 63% train / 12% validation / 25% test.

- **StackOverflow**:
  - **Role**: Represents a dataset where labels correspond to technical topics (20 programming languages/technologies) rather than user intents, offering a contrast in structure.
  - **Size**: Contains 20,000 utterances, split 60% train / 10% validation / 30% test.

These datasets provide a spectrum of granularities, domain specificities, and underlying structures against which the properties of our newly created corporate email intent dataset, such as cluster separability and cohesion, can be assessed.

4.2 ANALYSIS AND EVALUATION PROCEDURES

This section details the specific procedures implemented to characterize the labeled corporate email intent dataset and compare its properties against established benchmark datasets, using the evaluation strategies defined in Section 3.3. To facilitate reproducibility of these procedures, particularly those applied to publicly available data, the code implementing the dataset characterization workflows and is available at: `https://github.com/emailintents/benchmark_experiments/`. Note that due to licensing restrictions, the primary labeled dataset derived from the Avocado corpus is not included in this repository.

4.2.1 *Labeled Dataset Characterization Procedure*

The procedures described here were executed to analyze the distributional, semantic, and lexical properties of the final labeled corporate email intent dataset (specified in Section 4.1.1) generated through the methodology in Chapter 3.

DATASET DISTRIBUTION ANALYSIS. Two distinct procedures were employed to analyze intent distributions: one characterizing the final labeled dataset resulting from the methodology, and another supplementary analysis aimed at approximating the underlying distribution within the original data partitions.

- **A) Distribution within the Final Labeled Dataset**: To visualize the frequency and skewness of intent labels within the final, curated dataset splits, the occurrences of each of the 54 in-scope intent labels were counted. These counts were then plotted as barcharts using both the Matplotlib and Seaborn libraries to illustrate the compositional characteristics of the curated dataset resulting from the applied methodology.

- **B) Approximation of Underlying Distribution via Random Sampling**: In order to gain insights into the likely distributions of intents within the original, uncurated data partitions and to assess taxonomy applicability on random data, a supplementary sampling analysis was conducted. From each of the original unlabeled training, validation and test partitions, **n=100** data points (utterances) were randomly sampled without replacement. These 300 sampled utterances were manually labeled by the author according to the final 54-class intent taxonomy. Utterances that could not be confidently assigned to an existing category due to ambiguity, lack of context, or representing a potentially new intent were assigned a new intent label that best describes the underlying goal of the utterance. Subsequently, the frequencies of the assigned labels within each 100-sample set were calculated. These frequencies were visualized as barcharts using Matplotlib/Seaborn to approximate the underlying intent distribution within each original partition.

To visualize the frequency and skewness of intent labels, histograms were generated for the training, validation, and test splits separately. Label counts for each intent class within each split were computed, and these counts were plotted using the Matplotlib library. However, as our methodology is more about curating a dataset, a separate experiment is being conducted to approximate the true intent distribution. For this we sampled n=100 datapoints from each respective dataset partition to gain insights into the true underlying distribution of intents within our analyzed subset.

SEMANTIC STRUCTURE VISUALIZATION (UMAP). To visually inspect the semantic relationships and separability of intent classes, the UMAP technique [McI+18] was employed. First, sentence embeddings for all labeled utterances were generated using the pre-trained SBERT model `all-MiniLM-L6-v2` from the Hugging Face Sentence Transformers library [1]. UMAP, implemented via the `umap-learn` library [2], was then applied to these 384-dimensional embeddings to project them into a 2-dimensional space suitable for plotting. Key UMAP parameters were set as follows: `n_neighbors=30`, `min_dist=0.3`, `n_components=2`, and importantly, `metric='cosine'` to reflect the similarity measure most appropriate for these embeddings. Class centroids were also computed by averaging the SBERT embeddings for all utterances within each intent class and projected using the same UMAP transformation for visualization.

LEXICAL CHARACTERISTIC ANALYSIS. To identify dominant terms associated with each intent class, a lexical analysis was performed using the spaCy library [3], using the model `en_core_web_sm`. For each intent class, the corresponding utterances were processed by: (1) Tokenizing the text, (2) Performing Part-of-Speech (POS) tagging and lemmatization, (3) Removing standard English stopwords, (4) Counting the frequency of the remaining noun and verb lemmas. The top 3 most frequent noun lemmas and verb lemmas for each class were extracted.

1 https://huggingface.co/sentence-transformers
2 https://pypi.org/project/umap-learn/
3 https://spacy.io/

QUANTITATIVE CLUSTER QUALITY ASSESSMENT. To quantitatively assess the cohesion and separation of the intent classes within the embedding space, several metrics were calculated based on the SBERT embeddings (`all-MiniLM-L6-v2`).

- **Intra-Class Similarity**: The cohesiveness within each defined intent class was quantified by calculating the average pairwise cosine similarity among its constituent utterances. This involved the following steps:
  - All utterance embeddings were stacked into a numerical matrix using NumPy [4] (`np.vstack`).
  - A comprehensive pairwise cosine similarity matrix, comparing every utterance embedding against every other, was computed using Scikit-learn's `cosine_similarity` function.
  - For each unique intent label, the sub-matrix corresponding to similarities *only between utterances belonging to that specific label* was extracted using Pandas [5] DataFrame indexing.
  - To ensure the average was not inflated by self-similarity (which is always 1), the diagonal elements of each sub-matrix were disregarded (conceptually set to `NaN`).
  - The mean (`np.nanmean`) and variance (`np.nanvar`) of the remaning off-diagonal similarity values were computed for each intent class, representing the **Average Intra-Class Similarity** and its **Variance** respectively.
  - Finally, the overall mean of the per-class Average Intra-Class Similarities was calculated to provide a single summary statistic for the dataset's internal cohesion.
- **Inter-Class Similarity**: To estimate the semantic proximity *between different* intent classes, the average pairwise cosine similarity was computed for every unique pair of distinct intent classes. The procedure was as follows:
  - All unique pairs of distinct intent labels were generated using Python's `itertools.combinations`.
  - For each pair of intents (e.g., `intent_a` and `intent_b`), the corresponding sets of utterance embeddings were extracted and stacked into separate numerical matrices (`emb_a`,`emb_b`) using NumPy (`np.stack`).
  - The Scikit-learn `cosine_similarity` function was then used to compute the similarities between *all* utterances in `embd_a` and *all* utterances in `embd_b`, resulting in a matrix of cross-class similarity scores.
  - The overall mean (`np.mean`) of this cross-class similarity matrix was calculated, yielding a single value representing the average semantic similarity between `intent_a` and `intent_b`.

4 https://numpy.org/
5 https://pandas.pydata.org/

  - This process was repeated for all unique pairs of intents, and the results were organized into a symmetric similarity matrix using Pandas, capturing the average similarity between all distinct intent classes.

- **Silhouette Score & DBI**: Overall dataset cluster quality was assessed using the Silhouette Score and DBI. While the implementation of the Silhouette Score already These were calculated using implementations from the Scikit-learn library [6] (`silhouette_score`, `davies_bouldin_score`), critically adapting them to use *cosine_distance* (defined as $1 - cosine\_similarity$) as the distance metric. These metrics were computed using the assigned labels for our labeled dataset.

4.2.2 *Comparison with Benchmark Datasets Procedure*

To establish a comparative baseline for understanding the structural characteristics of the newly created labeled corporate email dataset, the core quantitative analysis procedures outlined in Section 4.2.1 were systematically applied to each of the benchmark intent datasets specified in Section 4.1.2 (SNIPS, BANKING77, CLINC150, and StackOverflow). The goal was to ensure that any observed differences in metrics reported in Chapter 5 reflect inherent dataset properties rather than variations in the analytical methodology. A fundamental aspect of this procedure was the consistent application of the *same* tools, models, and metric calculations across all datasets. Thus the generation of embeddings and measured metrics was conducted the exact same way as described for the newly created labeled corporate email dataset. This yields a set of comparable structural metrics (overall intra-similarity, inter-similarity, Silhouette Score, DBI) for each benchmark dataset, which are presented alongside those of the primary dataset for comparative analysis.

4.2.3 *Label Quality Assessment via Cleanlab Procedure*

To quantitatively evaluate assess the quality of the labels assigned during the intent discovery and labeling process (described in Chapter 3) and to identify instances and patterns of potential label noise within the created dataset, the Confident Learning framework, as implemented in the `cleanlab` Python library [7], was employed. Given that the labeling process was applied uniformly and the resulting labels across all predefined splits (train, validation, test) were considered potentially noisy "pseudo-labels" rather than absolute ground truth, this analysis was performed on the **entire combined labeled dataset**. This approach allows for a comprehensive identification of conflicts across all available labeled data points.

GENERATING OUT-OF-SAMPLE PREDICTED PROBABILITIES. Confident Learning requires predicted probabilities for each data point that are generated Out-of-Sample

6 https://scikit-learn.org
7 https://github.com/cleanlab/cleanlab

(OOS) (i.e., from a model not trained on that specific data point). To achieve this, a **stratified k-fold cross-validation** procedure was implemented:

- **Splitting**: The combined dataset was divided into $k = 5$ folds using stratified splitting (`sklearn.model_selection.StratifiedKFold`) to maintain the proportional representation of each intent class within each fold, which is crucial due to the observed label imbalance (Section 5.1.2).
- **Iteration**: A loop iterated 5 times, with each fold serving once as the held-out validation set and the remaining $k - 1$ (i.e., 4) folds serving as the training set for that iteration.
- **Fold-Specific Body Fine-Tuning**: Within each iteration:
  - A fresh base Sentence Transformer model was loaded using `SetFitModel.from_pretrained`.
  - The `SetFitTrainer` was configured with the training texts and labels for the current k-1 folds.
  - The `trainer.train()` method was executed, primarily to perform **contrastive fine-tuning of the SBERT model body** based on the training data of this specific fold. Key SetFit training parameters included `num_iterations=20` and `batch_size=16`. *(Note: The head trained internally by SetFitTrainer during this step is temporary and not used for the final probability prediction).*
  - The fine-tuned SBERT body (`trainer.model.model_body`) specific to this fold was extracted.
- **Fold-Specific Embedding Generation**: Using the **fine-tuned body** obtained in the previous step, sentence embeddings were generated for *both* the training utterances (k-1 folds) and the held-out validation utterances (1 fold) for the current iteration.
- **Fold-Specific Classifier Head Training**: A **separate Logistic Regression classifier** (`sklearn.linear_model.LogisticRegression`) was trained exclusively on the *training embeddings* and corresponding *training labels* for this iteration. Recognizing the significant class imbalance inherent in real-world communication data and confirmed during preliminary analysis of our dataset, the classifier was configured with the `class_weight='balanced'` parameter to mitigate potential bias towards majority classes (the detailed distribution analysis is presented in Section 5.1.2). Other parameters included `max_iter=2000` and `solver='liblinear'`.
- **Out-of-Sample probability Prediction**: The trained Logistic Regression head was then used to predict class probabilities (`predict_proba`) for the embeddings of the *held-out validation fold*.
- **Probability Aggregation**: These predicted probabilities for the validation fold were stored. By mapping them back to the original indicies of the utterances, a complete matrix (`oof_preds`) containing OOS predicted probabilities for every utterance in the entire dataset was assembled after all 5 folds were processed.

CLEANLAB EXECUTION. The generated matrix of OOS predicted probabilities and the corresponding array of original (potentially noisy) labels assigned during the dataset creation process were provided as input to the `cleanlab.filter.find_label_issues` function. This function identified instances suspected of having incorrect labels based on the Confident Learning algorithms.

VISUALIZATION SETUP.

- **Conflict Matrix**: A confusion matrix, termed the "Label Conflict Heatmap" (Figure 5.7), was generated using Matplotlib/Seaborn. This matrix cross-tabulates the original assigned labels against the labels predicted with the highest OOS probability *specifically for those instances flagged by Cleanlab as potential issues*. This highlights the primary sources of confusion.
- **Network Graph**: A directed network graph (Figure 5.8) was constructed using the NetworkX library [8] and visualized using Matplotlib/Seaborn with a force-directed layout (e.g., `spring_layout`). Nodes represent the intent classes. Directed edges from class A to class B indicate Cleanlab identified instances originally labeled 'A' but predicted as 'B'. Edge thickness was scaled proportionally to the count of such conflicts. Node size was scaled based on the total number of off-diagonal conflicts involving that class (both incoming and outgoing) as identified by Cleanlab.

4.2.4 *TEXTOIR Experiment Setup*

To assess the performance of representative state-of-the-art open-set intent detection and semi-supervised intent discovery algorithms when applied to the challenging characteristics (skew, semantic overlap) of the newly created labeled corporate email dataset, experiments were conducted utilizing the standardized TEXTOIR framework [Zha+21c]. This allows for evaluating how well these established methods generalize to the nuances of this specific dataset and provides insights into their robustness under varying conditions of data availability and class knowledge.

ALGORITHMS SELECTED. On the side of intent detection, the ADB [ZXL21] method was chosen as a representative state-of-the-art algorithm for this task. For the intent discovery task, Deep Aligned Clustering [Zha+21b] was selected to evaluate performance in discovering intents.

DATASET CONFIGURATIONS. Recognizing the significant class imbalance in the final labeled dataset, experiments were conducted using two distinct dataset splitting strategies to analyze potential impacts:

- **Original Skewed Splits**: Utilizing the standard train (4804 samples), validation (1229 samples), and test (752 samples) sets as defined in Section 4.1.1. This configuration preserves the skew resulting from the data creation methodology.

8 https://networkx.org/

- **Stratified Splits**: Employing newly generated train, validation, and test splits (using a split of 90%/5%/5%) created via stratified sampling from the *entire* combined labeled dataset. This configuration ensures proportional class representation across all partitions, providing a comparison point against the original setup.

TEXTOIR PARAMETER VARIATIONS. Key parameters within the TEXTOIR framework were systematically varied to simulate different realistic scenarios and assess model robustness:

- **KIR**: This parameter was varied (across the values [0.25, 0.50, 0.75]) to simulate scenarios where only a fraction of the total intent classes (54 in this case) are known during training. The model's task is to correctly classify instances of known classes while identifying instances belonging to the remaining (temporarily held-out) classes as 'unknown' (OOD).
- **LR**: This parameter controls the fraction of the available *training data* for the *known classes* that is actually provided with labels during training. It was varied (e.g., across values such as [0.1 0.2 0.3 ... 1.0]) to simulate low-resource learning scenarios and assess how much labeled data is sufficient for effective model training on this dataset.

EVALUATION METRICS. Performance for these TEXTOIR experiments was measured using standard evaluation metrics suitable for the respective tasks:

- For **open intent detection** (evaluated using ADB), performance was assessed using standard classification metrics such as Accuracy and F1-scores (calculated separately for known and unknown/OOD classes).
- For **semi-supervised intent discovery** (evaluated using Deep Aligned Clustering), the quality of the generated clusters was measured using **NMI** and **ARI**, as defined in Section 3.3.4. These metrics quantify the alignment between the discovered clusters and the ground truth intent labels.

5

RESULTS

This chapter presents the empirical findings derived from the application of the methodologies and experimental procedures detailed in Chapters 3 and 4. The primary focus is on characterizing the newly created labeled corporate email intent dataset, evaluating its properties in the context of established benchmarks, assessing its label quality, and reporting the performance of standard intent analysis algorithms when applied to it.

The chapter begins with a multi-faceted analysis of the labeled dataset generated in this work (Section 5.1). This includes a qualitative overview of the discovered intent themes, an examination of intent distributions (both within the final curated dataset and approximated from random sampling), visualization of the semantic structure using UMAP, lexical analysis of characteristic terms, and quantitative metrics assessing cluster cohesion and separation.

Subsequently, Section 5.2 presents the results of applying similar characterization analyses to four standard benchmark intent datasets (SNIPS, BANKING77, CLINC150, StackOverflow) to establish a comparative baseline.

Building on these individual analyses, Section 5.3 provides a comparative summary, directly contrasting the key structural metrics of the generated dataset against the benchmarks.

Following the dataset characterizations, Section 5.4 reports the performance outcomes obtained by running selected state-of-the-art open-set intent detection (ADB) and discovery (Deep Aligned) algorithms from the TEXTOIR framework on our dataset under various configurations.

Finally, Section 5.5 details the findings from the label quality assessment conducted using Cleanlab, quantifying potential inconsistencies and highlighting specific patterns of label confusion within the generated dataset.

5.1 LABELED DATASET CHARACTERISTICS

This section presents a comprehensive analysis of the labeled corporate email intent dataset generated through the methodology outlined in Chapter 3. The characteristics of this dataset are examined from multiple perspectives to understand its composition, structure, and properties. The following subsections detail the qualitative themes discovered during taxonomy development, the quantitative distribution of intents within the final dataset and approximations from random sampling, the semantic structure visualized via dimensionality reduction, a lexical analysis of intent-specific terminology, and finally, quantitative metrics assessing overall cluster quality.

5.1.1 *Qualitative Overview of Discovered Intent Categories*

Qualitative inspection of the intent clusters generated via the methodology described in Chapter 3 revealed several recurring thematic categories reflecting typical workplace communication needs.

A central group of intents pertains to IT-related operations, including utterances concerning file transfers, requests for access to systems or infrastructure, inquiries about login credentials, and bug/issue management. Another major category involves project coordination and task management, encompassing requests to be kept informed , offers of support, and inquiries about task procedures.

A broad class of document and information exchange intents was also identified, ranging from requests to send specific documents (resumes, reports) and timesheet submissions to utterances related to reviewing or signing documents and requesting contact details. This category also included informative messages notifying recipients of attachments. Furthermore, a distinct set of intents centered on meeting logistics and scheduling, such as setting up meetings, checking availability, or inquiring about time/location. Finally, several administrative or specialized intents like hotel reservations or confidentiality requests were observed.

Analysis of cluster granularity revealed that while larger clusters often represented these core themes, smaller clusters (e.g., those with five or fewer utterances) typically contained highly specific phrasings or contextual nuances of already identified intents, rather than representing entirely novel categories.

Variations in cluster coherence were also noted. Some clusters, like those for `offer_assistance`, exhibited high homogeneity, with utterances consistently conveying the same core meaning. Others, such as those related to `request_contact`, were more heterogeneous, sometimes combining requests for being contacted with requests for others' contact information. Examples of multi-intent utterances within single conceptual clusters were encountered (e.g., simultaneously requesting to print and sign a document).

Additionally, certain utterances proved difficult to assign to a single intent category due to a lack of surrounding conversational context (which was intentionally excluded in the sentence-level analysis). Instances were also found where clusters grouped utterances with opposite meanings, particularly involving negation (e.g., requests to forward information clustered with requests explicitly asking *not* to forward a specific piece of information).

Based on this inspection and analysis of recurring patterns, the final intent taxonomy comprising 54 in-scope categories was defined (see Table A.4).

5.1.2 *Intent Distribution in the final labeled dataset*

This section characterizes the frequency distribution of the 54 in-scope intent labels within the final training, validation, and test splits. These splits constitute the dataset artifact generated via the discovery and iterative refinement methodology detailed in Chapter 3.

It is crucial to emphasize that this distribution is a direct outcome of that methodology - reflecting the intents successfully identified and populated with sufficient examples through targeted filtering, LLM-based clustering, and iterative expansion. **Therefore, the observed frequencies presented here characterize the specific dataset used for subsequent analyses and model evaluations in this thesis, but they do not necessarily represent the true underlying frequency or prevalence of these intents within the original, uncurated Avocado email corpus.** Understanding this specific distribution is vital, however, as it highlights the inherent skew and composition of the data artifact upon which further experiments are based.

A separate analysis presented in the next section (Section 5.1.3) uses random sampling to provide an approximation of the underlying intent distribution within the original data partitions, offering a valuable contrast to the curated dataset's composition.

Examining the composition of this final labeled dataset reveals the following:

- **Most Frequent Intent**: The `offer_assistance` category contains the highest number of labeled utterances across all splits.
- **Substantially Represented Intents**: A significant portion of the dataset comprises intents related to coordination, communication, and scheduling.
- **Notably Represented Intents**: Categories associated with document handling and routine administrative tasks are also well-represented.
- **Less Frequent Intents**: Several intents representing more specific functions, such as IT issue management appear with lower frequencies.
- **Rarest Intents**: Intents observed with the fewest examples in the labeled splits include social intents and more specific requests.

In summary, the final labeled dataset used for subsequent evaluations is compositionally dominated by intents reflecting core operational communications (assistance, meetings, calls, document exchange) typical of the analyzed corporate environment, a direct outcome of the applied data curation methodology.

5.1.3 *Approximate Intent Distributions found in the respective dataset splits*

To estimate the underlying distribution of the intents within the original, uncurated data partitions and to assess the direct applicability of the final taxonomy, a supplementary analysis was conducted by randomly sampling and manually labeling 100 utterances from each original split.

- **Domincance of General Intents**: The intent `offer_assistance` was found to be overwhelmingly dominant in the random samples across all three splits, constituting a significantly larger proportion than any other category.
- **Presence of Core Communication Intents**: Other intents consistently appearing with moderate frequency in the random samples included standard administrative tasks, dealing with calls, requests for documents and the planning of appointments.
- **Sparsity of Specific Intents**: A key finding was the **extreme rarity or complete absence** of the vast majority of the 54 defined in-scope intents within these 100-utterance random samples. Most specific task-related or administrative intents occurred only one or twice, if at all, in any given sample.
- **Labeling Challenges Observed**: This manual labeling exercise on random samples also highlighted significant challenges not fully reflected in the curated dataset. Some of the sampled utterances were difficult or impossible to assign to an existing category due to ambiguity, lack of context, or high specificity overall. Other utterances, however, suggested potential new categories beyond the established taxonomy.

While the small sample size per split (n=100) limits precise quantification for rare intents, these results strongly indicate that **most fine-grained intents identified in this research are relatively sparse within the general email corpus**. The underlying distribution appears heavily skewed towards a few very common communicative functions like offering assistance. This observation underscores the difference between the composition of the final, curated dataset (enriched for specific intents via the methodology) and the probable distribution in the raw data partitions, further justifying the targeted approach taken for dataset creation while also highlighting the challenge of achieving exhaustive intent coverage.

5.1.4 *Semantic Structure Analysis*

Figures Figure 5.1 and Figure 5.2 present the 2-dimensional UMAP projections derived from the SBERT embeddings of the labeled dataset utterances and their corresponding centroids, respectively, visualizing semantic relationships. For better clarity, the plots are separated.

The visualization in Figure 5.1 indicates considerable overlap among many intent classes within the 2D semantic space, suggesting a lack of clear separation for a large

portion of the categories. However, some exceptions are observable; for instance, the cluster corresponding to `request_login_credentials` (blueish cluster, bottom left quadrant) appears relatively distinct from the main grouping. Similarly, the cluster for `mark_calendar` (orange cluster, bottom right quadrant) shows some separation.

Proximity between related intents is also visible. For example, intents related to meeting scheduling, such as `request_reschedule`, `propose_meeting`, and `request_meeting`, are located near each other in the bottom right quadrant, close to the `mark_calendar` cluster.

Figure Figure 5.2 displays the calculated centroids for each class, illustrating their central position within the embedding space relative to each other. The plot shows variation in the spatial distribution of intent categories, with some appearing relatively tightly grouped while others are more dispersed. Quantitative analysis of cluster separation and cohesion is presented in Section 5.1.6.

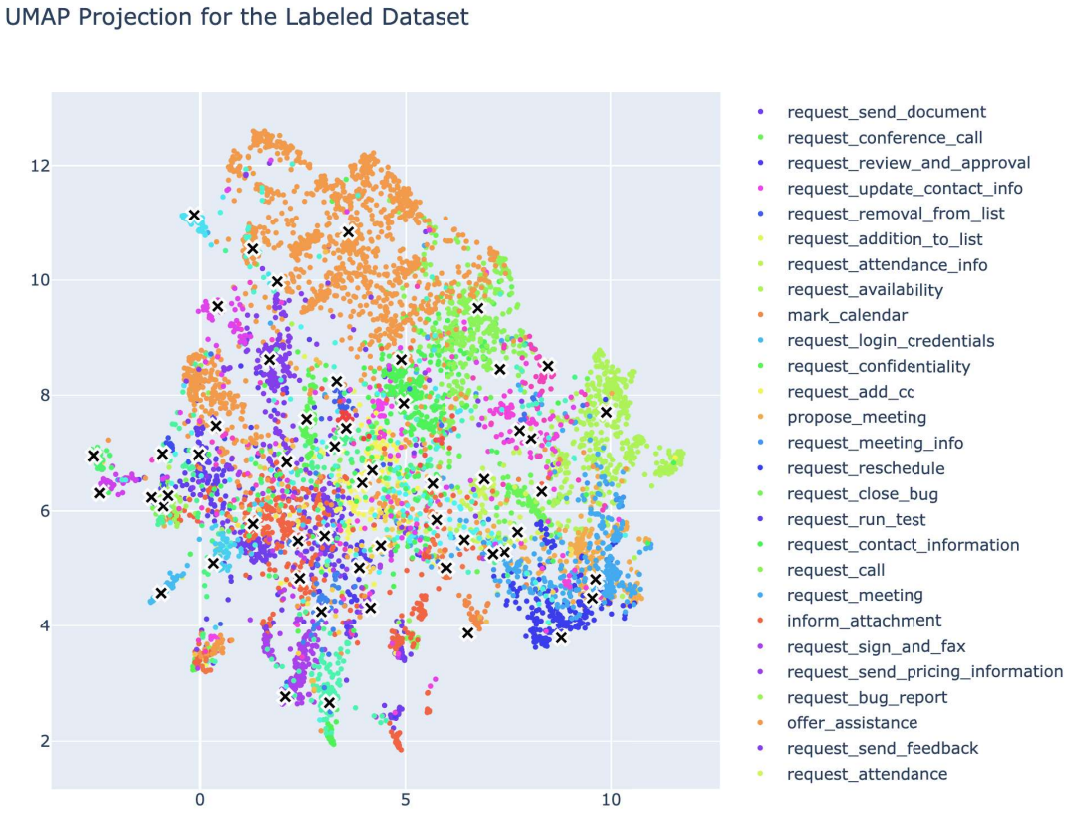

Figure 5.1: UMAP visualization of the embeddings produced for the labeled dataset. Best viewed in color.

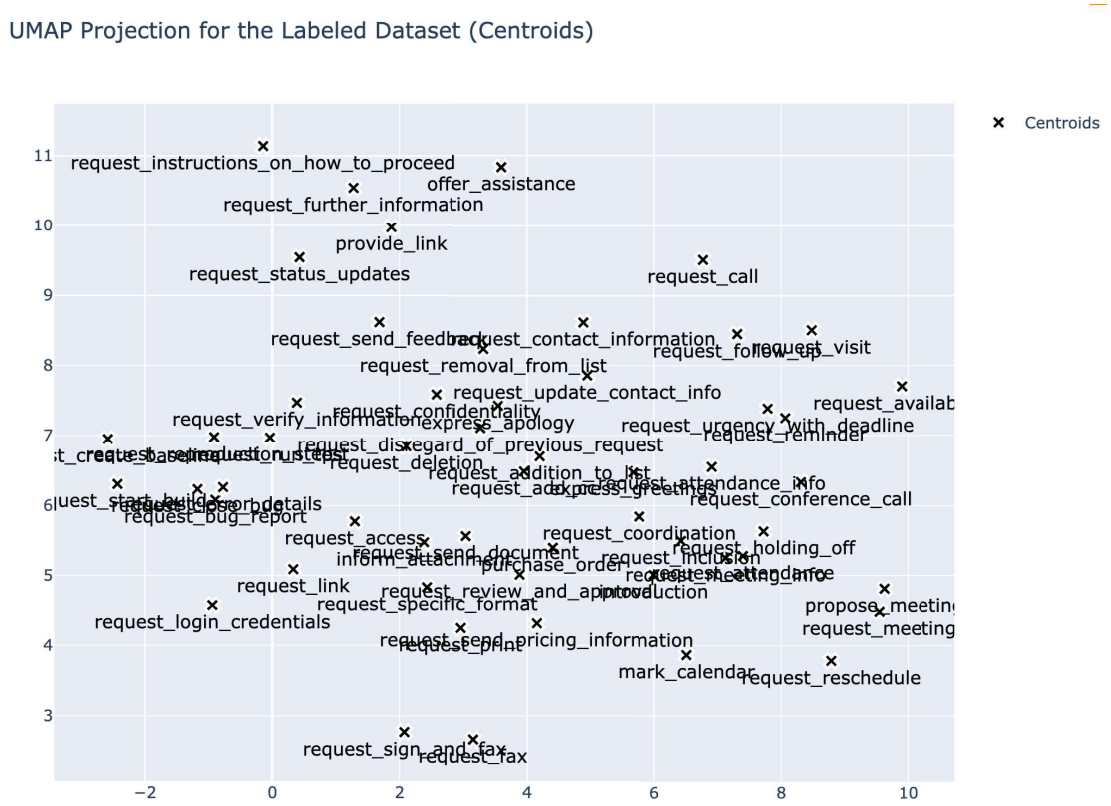

Figure 5.2: UMAP visualization of the centroids for the embeddings of a respective class.

5.1.5 *Lexical Analysis*

The lexical analysis provides insight into the characteristic vocabulary of each intent.

Some intents, such as `purchase_order` and `request_login_credentials`, utilize terminology with minimal overlap compared to other intents.

Conversely, several classes exhibit significant lexical similarity, reflecting related thematic content. For example, `request_meeting` and `request_meeting_info` share core vocabulary related to meeting arrangements. Similar overlaps can be observed between other functionally related intents.

5.1.6 *Quantitative Cluster Quality*

The internal cohesion of each intent class within the SBERT embedding space was quantified by calculating the average intra-class cosine similarity, with results summarized in Table 5.1. It reveals a range of cohesiveness across the different intent categories. The highest average intra-class similarity was observed for the intent `request_start_build` (0.446). High values were also recorded for `offer_assistance` (0.383), `request_call` (0.359), `request_fax` and `request_instructions_on_how_to_proceed` with both being at 0.357.

Conversely, several intents displayed lower internal similarity. The `request_confidentiality` intent showed an average intra-class similarity of 0.203, while

the `oos` category, by design containing diverse utterances not belonging to defined intents, had an average similarity of 0.194. The lowest measured average intra-class similarity was for the `request_verify_information` intent, at 0.184.

The overall mean of the average intra-class similarities, calculated across all 55 intent classes, including the oos intent class, was 0.265.

Table 5.1: Top and bottom 5 intents by average intra-class cosine similarity in the Labeled dataset.

Intent	Avg Intra Similarity	Variance
request_start_build	0.446	0.049
offer_assistance	0.383	0.029
request_call	0.359	0.037
request_fax	0.357	0.042
request_instructions_on_how_to_proceed	0.337	0.043
...	...	...
request_confidentiality	0.203	0.020
request_run_test	0.202	0.021
oos	0.194	0.016
provide_link	0.192	0.021
request_verify_information	0.184	0.017

The semantic proximity *between* different intent classes was assessed by calculating the average pairwise cosine similarity for all unique pairs of distinct intents. The pairs exhibiting the highest similarities are presented in Table 5.2.

As displayed, the highest measured average inter-class similarity (0.429) occurred between `request_sign_fax` and `request_fax`. Other pairs with notably high similarity scores include `request_close_bug` and `request_bug_report`, reflecting their shared focus on bug management processes.

Significant similarity was also observed among several meeting-related intents. For instance, the pairs `request_reschedule`, `request_meeting` (similarity: 0.408), and `request_meeting_info`, `request_meeting` (similarity: 0.383) all demonstrate considerable semantic overlap related to meeting logistics.

Finally, the overall quality of the intent class structure in the embedding space was assessed using global clustering metrics, calculated with cosine distance as described in Section 4.2. The **Silhouette Score for the labeled dataset was 0.057** and the **DBI was 3.257**.

5.2 BENCHMARK DATASET CHARACTERISTICS

To provide a comparative context for the labeled corporate email dataset presented in Section 5.1, this section details the results of applying the same analysis precedures (outlined in Section 4.2.1) to four established benchmark datasets: SNIPS, BANKING77, CLINC150, and StackOverflow. The findings presented here for each benchmark, cover-

Table 5.2: Pairs of intents with their respective cosine similarities

Intent 1	Intent 2	Cosine Similarity
request_sign_fax	request_fax	0.429
request_close_bug	request_bug_report	0.422
request_reschedule	request_meeting	0.408
propose_meeting	request_reschedule	0.389
propose_meeting	request_meeting	0.387
request_meeting_info	request_meeting	0.383
request_bug_report	request_reproduction_steps	0.378
request_conference_call	request_meeting	0.371
request_attendance_info	request_attendance	0.368
request_meeting_info	request_attendance	0.361
request_contact_information	request_call	0.359
request_conference_call	request_reschedule	0.357
request_conference_call	request_call	0.347
request_attendance_info	request_meeting_info	0.346
request_start_build	request_create_baseline	0.342
request_close_bug	request_reproduction_steps	0.341
request_availability	request_meeting	0.341
request_meeting	request_attendance	0.339
request_conference_call	request_meeting_info	0.338
request_availability	propose_meeting	0.336
request_conference_call	request_attendance	0.333
request_conference_call	propose_meeting	0.333

ing semantic structure, lexical patterns, and quantitative cluster quality metrics, establish a baseline for understanding the relative properties of the dataset developed in this research, which are quantitatively compared in Section 5.3.

5.2.1 *SNIPS Dataset Analysis*

The analysis results from the SNIPS dataset [Cou+18], including visualizations and quantitative metrics, are presented in Figure 5.3 and Table 5.3.

Overall structural metrics calculated for SNIPS (7 intents) using the procedures outlined in Chapter 4 include a include a **Silhouette Score of 0.1508** and a **DBI of 2.697**.

Detailed metrics within the dataset are as follows:

- **Semantic Structure (UMAP)**: The UMAP projection in Figure 5.3 visualizes the utterance embeddings and class centroids. Observable semantic overlap exists between certain classes, notably `AddToPlaylist` and `PlayMusic`, and also among `SearchCreativeWork`, `SearchCreativeWork` and `RateBook`. In contrast, the

`GetWeather` and `BookRestaurant` clusters appear visually distinct from other categories. Some outliers are present, for instance, the utterance *"add nana tanimura to a sudden rainstorm"* (labeled `AddToPlaylist`) plots closer to the `GetWeather` cluster space in this 2D projection.

- **Lexical Analysis**: Table 5.3a lists the top 3 noun and verb lemmas for each of the 7 intent classes. Noun-verb pairs often correspond directly to the intent label, such as "playlist"/"add" for `AddToPlaylist` and "restaurant"/"book" for `BookRestaurant`. Some potentially verbal terms like "rate" in `RateBook` were identified as top nouns by the POS tagger.
- **Intra-Class Similarity**: The average intra-class cosine similarities for the intents are shown in Table 5.3b, ranging from a minimum of 0.154 for `SearchCreativeWork` to a maximum of 0.273 for `SearchScreeningEvent.`
- **Inter-Class Similarity**: The matrix of average inter-class cosine similarities is presented in Table 5.3c. The highest similarity (0.265) was observed between `PlayMusic` and `AddToPlaylist`. Other notable similarities include 0.147 between `SearchScreeningEvent` and `SearchCreativeWork`, and 0.126 between `SearchScreeningEvent` and `BookRestaurant`.
- **Similar Utterance Examples**: Table 5.3d provides examples of specific utterance pairs from different intent classes that exhibit high cosine similarity, such as an utterance for `PlayMusic` and `AddToPlaylist` having a similarity of 0.879.

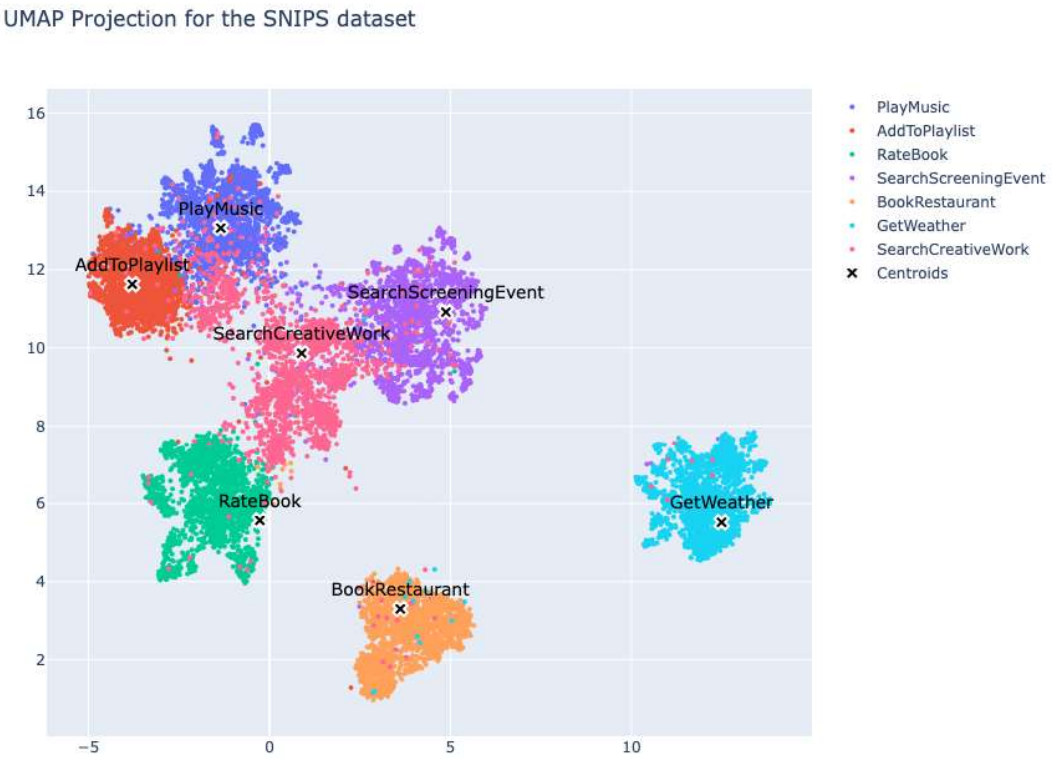

Figure 5.3: UMAP visualization of the embeddings produced for the SNIPS dataset. Cluster centroids with the respective labels are included. Best viewed in color.

Table 5.3: Analysis of the SNIPS dataset.

Label	Top Nouns	Top Verbs
AddToPlaylist	[playlist, song, tune]	[add, want, call]
BookRestaurant	[book, restaurant, table]	[need, book, want]
GetWeather	[weather, forecast, park]	[s, go, tell]
PlayMusic	[music, song, track]	[play, hear, want]
RateBook	[star, point, rate]	[rate, want, get]
SearchCreativeWork	[tv, game, saga]	[find, call, look]
SearchScreeningEvent	[movie, schedule, theatre]	[play, find, animate]

(a) Top 3 nouns and verbs per intent category in the SNIPS dataset.

Intent	Avg. Intra Similarity	Variance
SearchScreeningEvent	0.273	0.037
BookRestaurant	0.269	0.025
AddToPlaylist	0.265	0.026
RateBook	0.248	0.028
PlayMusic	0.247	0.019
GetWeather	0.226	0.023
SearchCreativeWork	0.154	0.011

(b) Average intra-class cosine similarity and variance for SNIPS intents, sorted in descending order.

	PlayMusic	AddToPlaylist	RateBook	SearchScreeningEvent	BookRestaurant	GetWeather	SearchCreativeWork
PlayMusic	1.000	0.265	0.086	0.122	0.085	0.023	0.145
AddToPlaylist	0.265	1.000	0.101	0.100	0.104	0.039	0.133
RateBook	0.086	0.101	1.000	0.084	0.111	0.058	0.101
SearchScreeningEvent	0.122	0.100	0.084	1.000	0.126	0.077	0.147
BookRestaurant	0.085	0.104	0.111	0.126	1.000	0.089	0.098
GetWeather	0.023	0.039	0.058	0.077	0.089	1.000	0.046
SearchCreativeWork	0.145	0.133	0.101	0.147	0.098	0.046	1.000

(c) Inter-cluster cosine similarity matrix between SNIPS intent categories.

Utterance 1 [Label 1]	Utterance 2 [Label 2]	Cosine Similarity
play my 70s smash hits playlist *[PlayMusic]*	add this track to the 70s smash hits playlist *[AddToPlaylist]*	0.879
what time is cabin fever: spring fever playing *[SearchScreeningEvent]*	play the cabin fever 2: spring fever saga *[SearchCreativeWork]*	0.831
play some house music *[PlayMusic]*	play the home is where the music is tv series *[SearchCreativeWork]*	0.743

(d) Examples of very similar utterances stemming from different intent categories.

5.2.2 *BANKING77 Dataset Analysis*

The analysis results for the BANKING77 dataset [Cas+20], featuring 77 fine-grained intents within the financial services domain, are presented in Figure 5.4 and Table 5.4.

The overall structural metrics calculated for this dataset are a **Silhouette Score of 0.1564** and a **DBI of 2.470**.

Detailed metrics observed within the dataset include:

- **Semantic Structure (UMAP)**: The UMAP projection in Figure 5.4 shows a fluid structure with considerable overlap between many of the 77 intent clusters. While significant mixing is apparent, some intents associated with more specialized terminology (e.g., those potentially involving "PIN", "fraud", "ATM") may occupy relatively more distinct regions with the projection.
- **Lexical Analysis**: Table 5.4a shows the top 3 noun and verb lemmas for selected BANKING77 intents. Recurring financial terms such as "card", "account", and "money" are frequent top nouns, while general-purpose verbs like "use","get","give" and "need" appear across multiple classes. Some intents feature more distinctive verbs, such as "verify", "withdraw", or "reject".
- **Intra-Class Similarity**: The average intra-class cosine similarities (Table 5.4b) demonstrate high cohesion for certain intents, reaching a maximum of 0.426 for `activate_my_card`. The lowest observed value was 0.247 for `country_support`.
- **Inter-Class Similarity**: Table 5.4c lists the top-5 most similar intent pairs based on average inter-class cosine similarity. The highest similarity (0.575) was measured between `why_verify_identity` and `verify_my_identity`. Other pairs with high similarity include `change_pin` and `get_physical_card` (0.571), and `getting_virtual_card` and `virtual_card_not_working` (0.570).
- **Similar Utterance Examples**: High similarity scores between specific utterances from different classes are exemplified in Table 5.4d, such as a pair from `why_verify_identity` and `verify_my_identity` achieving a similarity of 0.985.

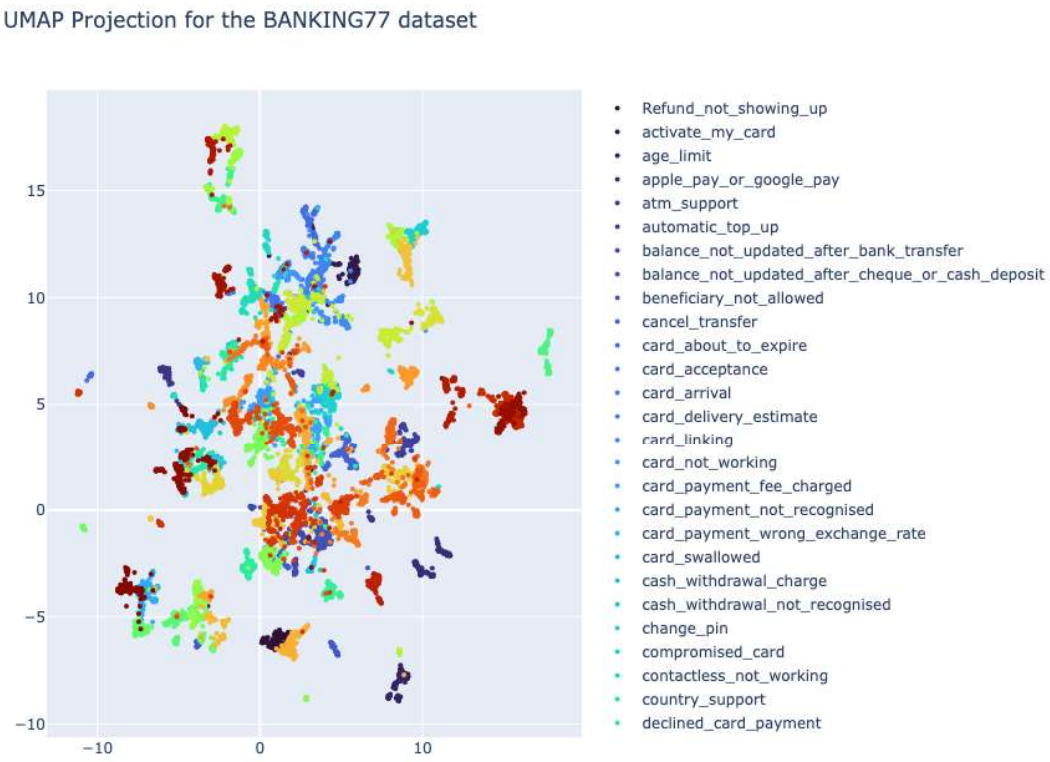

Figure 5.4: UMAP visualization of the embeddings produced for the BANKING77 dataset. Cluster centroids are excluded due to the high amount of intent classes. Best viewed in color.

Table 5.4: Analysis of the BANKING77 dataset.

Label	Top Nouns	Top Verbs
Refund_not_showing_up	[refund, statement, account]	[show, request, check]
activate_my_card	[card, activation, process]	[activate, need, get]
age_limit	[account, age, child]	[open, need, use]
apple_pay_or_google_pay	[apple, pay, watch]	[work, use, pay]
atm_support	[card, atm, money]	[use, accept, withdraw]
virtual_card_not_working	[card, work, payment]	[work, reject, use]
visa_or_mastercard	[card, visa, mastercard]	[choose, use, like]
why_verify_identity	[identity, account, verification]	[verify, need, use]
wrong_amount_of_cash_received	[cash, money, app]	[give, receive, withdraw]
wrong_exchange_rate_for_cash_withdrawal	[rate, exchange, cash]	[get, apply, withdraw]

(a) Top 3 nouns and verbs per intent category in the BANKING77 dataset.

Intent	Avg. Intra Similarity	Variance
activate_my_card	0.426	0.079
cash_withdrawal_charge	0.395	0.053
card_payment_wrong_exchange_rate	0.393	0.061
card_payment_fee_charged	0.393	0.055
reverted_card_payment?	0.387	0.047
...	...	...
lost_or_stolen_phone	0.268	0.045
edit_personal_details	0.264	0.051
age_limit	0.262	0.060
automatic_top_up	0.249	0.066
country_support	0.247	0.050

(b) Average intra-class cosine similarity and variance for BANKING77 intents, sorted in descending order.

Intent 1	Intent 2	Cosine Similarity
why_verify_identity	verify_my_identity	0.575
change_pin	get_physical_card	0.571
getting_virtual_card	virtual_card_not_working	0.570
wrong_exchange_rate_for_cash_withdrawal	card_payment_wrong_exchange_rate	0.550
get_disposable_virtual_card	getting_virtual_card	0.538

(c) Top-5 most similar query pairs based on cosine similarity.

Utterance 1 [Label 1]	Utterance 2 [Label 2]	Cosine Similarity
How to verify my identity *[why_verify_identity]*	How do I verify my identity? *[verify_my_identity]*	0.985
How do I change my card PIN? *[change_pin]*	How do I set-up my PIN for the new card? *[get_physical_card]*	0.890
What do I have to do to get my virtual card? *[getting_virtual_card]*	What do I have to do to get the virtual card to work? *[virtual_card_not_working]*	0.884

(d) Examples of very similar utterances stemming from different intent categories.

5.2.3 *CLINC150 Dataset Analysis*

The analysis results for the CLINC150 dataset [Lar+19], which includes 150 intents across 10 domains, are presented in Figure 5.5 and Table 5.5.

The dataset yielded the highest **Silhouette Score (0.220)** and the lowest **DBI (2.259)** among the evaluated benchmark datasets.

Detailed metrics observed include:

- **Semantic Structure (UMAP)**: The UMAP projection in Figure 5.5 shows a highly fragmented embeddding space with numerous small, relatively dense clusters scattered throughout. While many clusters appear distinct, indicating separation between numerous intents (e.g., potentially `are_you_a_bot`, `book_flight`), there are also denser regions where multiple intents seem to overlap or blend (e.g., potentially involving terms like "balance", "bill", "cancel").
- **Lexical Analysis**: The top noun and verb lemmas for selected intents are shown in Table 5.5a. Many classes exhibit distinct and indicative terminology, such as "block"/"freeze"/"lock" for `account_blocked` and "work"/"employ"/"know" for `who_do_you_work_for`, supporting the visual observation of distinct clusters for many intents.
- **Intra-Class Similarity**: Average intra-class cosine similarities (Table 5.5b) varied significantly, ranging from 0.081 for the `definition` intent up to 0.374 for `oil_change_how`.
- **Inter-Class Similarity**: The top-5 most similar intent pairs are listed in Table 5.5c. The highest similarity observed was 0.589 between `oil_change_when` and `oil_change_how`. Other high-similarity pairs include `credit_score` and `improve_credit_score` (0.543), and `report_lost_card` and `damaged_card` (0.512).
- **Similar Utterance Examples**: Table 5.5d presents examples of high-similarity utterance pairs from the classes exhibiting the highest inter-class similarity.

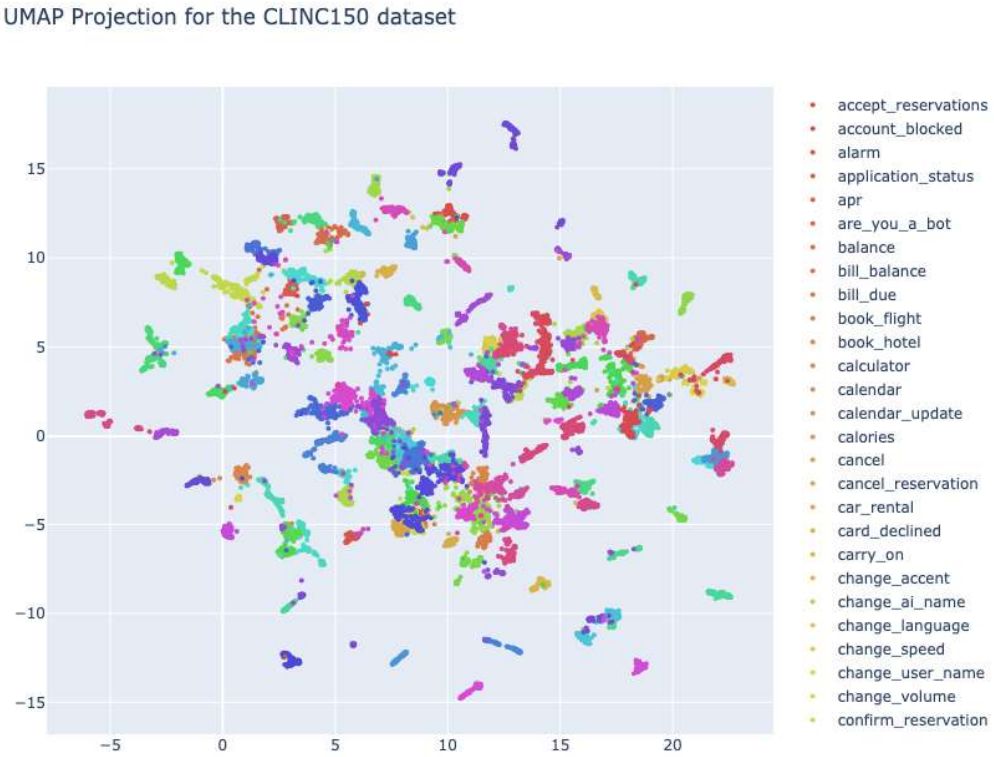

Figure 5.5: UMAP visualization of the embeddings produced for the CLINC150 dataset. Cluster centroids are excluded due to the high amount of intent classes. Best viewed in color.

Table 5.5: Analysis of the CLINC150 dataset.

Label	Top Nouns	Top Verbs
accept_reservations	[reservation, applebee, burger]	[take, know, tell]
account_blocked	[account, bank, hold]	[block, freeze, lock]
alarm	[alarm, tomorrow, pm]	[set, need, create]
application_status	[application, card, credit]	[process, know, go]
apr	[card, apr, credit]	[tell, apr, know]
where_are_you_from	[place, home, origin]	[bear, come, tell]
whisper_mode	[whisper, mode, voice]	[switch, whisper, use]
who_do_you_work_for	[boss, person, employer]	[work, employ, know]
who_made_you	[company, design, ai]	[know, program, tell]
yes	[answer, statement, response]	[confirm, agree, want]

(a) Top 3 nouns and verbs per intent category in the CLINC150 dataset.

Intent	Avg. Intra Similarity	Variance
oil_change_how	0.374	0.127
report_lost_card	0.359	0.106
apr	0.354	0.107
alarm	0.351	0.088
credit_score	0.347	0.107
...	...	...
ingredient_substitution	0.151	0.027
spelling	0.145	0.022
calculator	0.139	0.021
smart_home	0.135	0.029
definition	0.081	0.013

(b) Top and bottom 5 intents by average intra-class cosine similarity in the CLINC150 dataset.

Intent 1	Intent 2	Cosine Similarity
oil_change_when	oil_change_how	0.589
credit_score	improve_credit_score	0.543
report_lost_card	damaged_card	0.512
credit_limit	credit_limit_change	0.491
meeting_schedule	schedule_meeting	0.482

(c) Top-5 most similar intent pairs based on cosine similarity in the CLINC150 dataset.

Utterance 1 [Label 1]	Utterance 2 [Label 2]	Cosine Similarity
do i need to change my oil *[oil_change_when]*	what do i need to change my oil *[oil_change_how]*	0.861
tell me the steps to getting my credit score *[credit_score]*	what are some steps to building my credit score *[improve_credit_score]*	0.865
i lost my card and need to report it *[report_lost_card]*	my card has been erased and i need to report it *[damage_card]*	0.882

(d) Examples of very similar utterances stemming from different intent categories.

5.2.4 *StackOverflow Dataset Analysis*

The analysis results for the StackOverflow dataset, comprising 20 categories related to technology topics, are presented in Figure 5.6 and Table 5.6.

Overall structural metrics calculated for this dataset include the lowest **Silhouette Score (0.129)** among the benchmarks, and a **DBI of 2.202**.

Detailed observations from the analysis include:

- **Semantic Structure (UMAP)**: The UMAP projection for StackOverflow utterance embeddings is shown in Figure 5.6. The visualization displays a relatively structured embedding space where several clusters corresponding to different programming languages, frameworks, or technologies are visibly distinct. For example, categories such as `svn`, `qt`, and `matlab` appear as relatively compact and separated clusters. Other categories, including `svn`, `qt`, and `matlab`, are positioned in more peripheral regions of the 2D projection. A denser region with more noticeable overlap between classes exists near the center, involving categories like `magento`, `wordpress`, `drupal`, and `sharepoint`. Despite this central mixing, many categories maintain distinct centroid locations (indicated by black "X"s).

- **Lexical Analysis**: Table 5.6a displays the top 3 noun and verb lemmas for selected categories. While general verbs like "use","create" and "run" appear frequently, the noun distribution often feature the category label itself as the most frequent term (e.g., "apache", "bash", "spring", "svn").

- **Intra-Class Similarity**: The average intra-class cosine similarities, shown in Table 5.6b, are generally low, raning from 0.031 for `sharepoint` to a maximum of 0.126 for `linq`.

- **Inter-Class Similarity**: Average inter-class similarities are also modest, as indicated in Table 5.6c. The highest observed similarity was 0.165 between `hibernate` and `spring`, followed by 0.138 between `scala` and `haskell`.

- **Similar Utterance Examples**: Table 5.6d presents examples of high-similarity utterance pairs from the classes exhibiting the highest inter-class similarity.

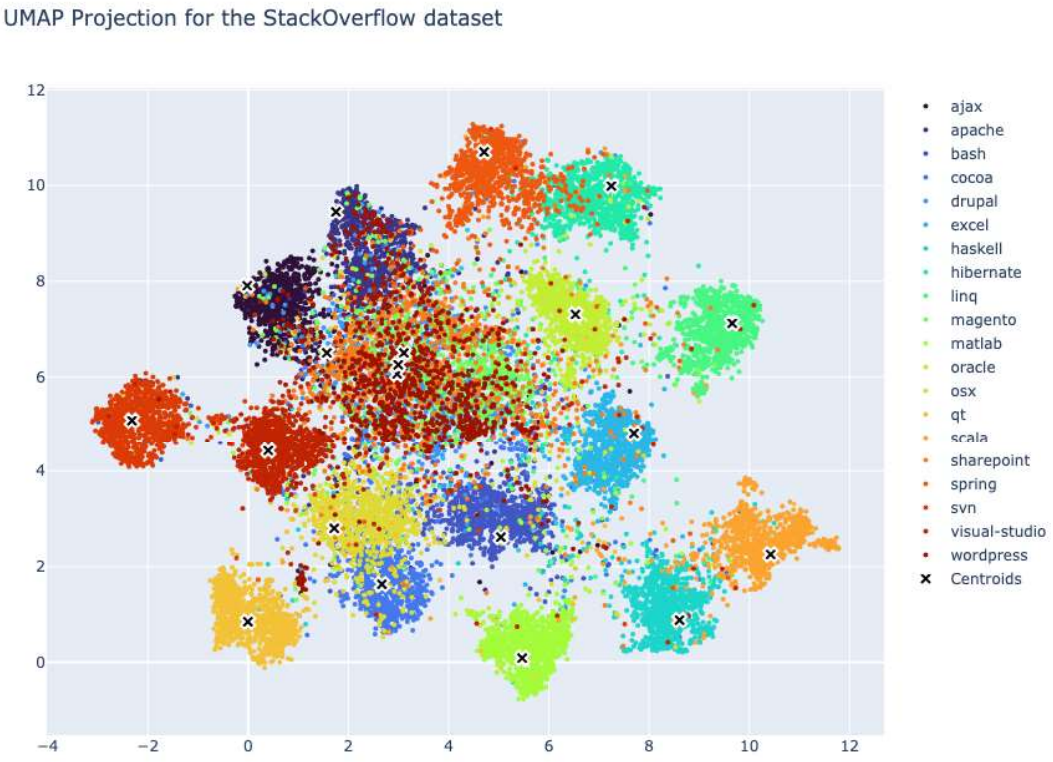

Figure 5.6: UMAP visualization of the embeddings produced for the StackOverflow dataset. Cluster centroids are included, but without the respective labels for better viewing. Best viewed in color.

Table 5.6: Analysis of the StackOverflow dataset.

Label	Top Nouns	Top Verbs
ajax	[request, page, javascript]	[work, return, update]
apache	[apache, file, server]	[redirect, rewrite, work]
bash	[bash, script, file]	[run, use, execute]
cocoa	[cocoa, application, way]	[create, use, add]
drupal	[drupal, view, node]	[create, add, change]
...	...	...
sharepoint	[sharepoint, list, web]	[create, add, sharepoint]
spring	[spring, bean, property]	[use, create, base]
svn	[svn, file, subversion]	[commit, use, work]
visual-studio	[project, file, studio]	[add, use, debug]
wordpress	[post, page, category]	[add, display, create]

(a) Top 3 nouns and verbs per intent category in the StackOverflow dataset.

Intent	Avg. Intra Similarity	Variance
linq	0.126	0.043
svn	0.115	0.035
hibernate	0.106	0.034
qt	0.099	0.031
scala	0.098	0.030
...	...	...
osx	0.054	0.013
wordpress	0.046	0.011
magento	0.038	0.009
drupal	0.037	0.010
sharepoint	0.031	0.009

(b) Average intra-class cosine similarity and variance for intents in the StackOverflow subset.

Intent 1	Intent 2	Cosine Similarity
hibernate	spring	0.165
scala	haskell	0.138
svn	visual-studio	0.097
matlab	excel	0.092
hibernate	linq	0.091

(c) Top-5 most similar intent pairs based on cosine similarity in the StackOverflow subset.

Utterance 1 [Label 1]	Utterance 2 [Label 2]	Cosine Similarity
Hibernate SessionFactory *[hibernate]*	SessionFactory in Hibernate *[spring]*	0.986
AnkhSVN vs VisualSVN *[svn]*	Which would you rather use: VisualSVN or AnkhSVN? *[visual-studio]*	0.955
Scala equivalent to Haskell Monads *[scala]*	Creating monads in haskell *[Creating monads in haskell]*	0.834

(d) Examples of very similar utterances stemming from different intent categories.

5.3 COMPARATIVE SUMMARY OF STRUCTURAL METRICS

To contextualize the structural properties of the created labeled corporate email dataset, its overall quantitative metrics were compared against those derived from established benchmark datasets. The results of this comparative analysis, including overall average intra-class similarity, overall average inter-class similarity, Silhouette Score and DBI, are summarized in Table 5.7.

Table 5.7: Comparison of the intent benchmark datasets with the labeled dataset.

Dataset	# Intents	Avg. Intra Sim (Max)	Avg. Intra Sim (Min)	Avg. Class Intra Sim	Top Inter Sim	Avg. Inter Sim	Silhouette Score	DBI
SNIPS	7	0.273 (SearchScreeningEvent)	0.154 (SearchCreativeWork)	0.240	0.265 (PlayMusic ↔ AddToPlaylist)	0.101	0.151	2.698
BANKING77	77	0.426 (activate_my_card)	0.247 (country_support)	0.332	0.575 (why_verify_identity ↔ verify_my_identity)	0.206	0.156	2.470
CLINC150	150	0.374 (oil_change_how)	0.081 (definition)	0.251	0.589 (oil_change_when ↔ oil_change_how)	0.082	0.220	2.259
StackOverflow	20	0.126 (linq)	0.031 (sharepoint)	0.077	0.165 (hibernate ↔ spring)	0.015	0.129	2.202
Ours	54	0.431 (request_start_build)	0.178 (provide_link)	0.265	0.429 (request_sign_and_fax ↔ request_fax)	0.172	0.057	3.257

Based on the values presented in Table 5.7:

- The **overall average intra-class similarity** of the created dataset (0.265) is comparable to CLINC150 (0.251) and SNIPS (0.240), but lower than BANKING77 (0.332) and significantly higher than StackOverflow (0.077).
- The **overall average inter-class similarity** of the created dataset (0.172) is notably higher than CLINC150 (0.082), SNIPS (0.101), and StackOverflow (0.015), but lower than BANKING77 (0.206).
- The **Silhouette Score** achieved by the created dataset (0.057) is the lowest among all evaluated datasets, with CLINC150 showing the highest score (0.220).
- Conversely, the **DBI** for the created dataset (3.257) is the highest (indicating poorer separation to this metric) compared to the benchmark datasets, where CLINC150 and StackOverflow showed the lowest DBI values (2.259 and 2.202, respectively).

5.4 TEXTOIR RESULTS

This section presents the performance results obtained by applying selected algorithms from the TEXTOIR framework to the labeled corporate email intent dataset under various experimental configurations, as detailed in Section 4.2.4. The aim is to assess how these standard methods handle the specific characteristics of the developed dataset

5.4.1 *Open Intent Detection (ADB Performance)*

The performance of the ADB algorithm [ZXL21] was evaluated on the labeled corporate email intent dataset using the TEXTOIR framework to assess its effectiveness in both classifying known intents and detecting OOD or unknown intents under varying conditions. The experiments explored the impact of the amount of training data (LR) and the proportion of known classes present during training (KIR).

IMPACT OF LABELED RATIO (LR). Table 5.8 presents the ADB performance on the `labeled_dataset_stratified` version (the version ensuring proportional class representation) while varying the LR from 0.3 to 1.0, keeping the KIR fixed at 1.0 (i.e., all 54 intent classes were considered "known"). As expected, there is a clear trend of improved performance as more labeled data becomes available. The F1-score for known classes increases steadily from 69.88% at LR=0.3 to 85.80% at LR=1.0. Similarly, overall Accuracy rises from 79.10% to 84.78%. Notably, performance is reasonably strong even with limited labeled data (e.g., achieving approximately 82% accuracy with only 50% of labeled data), though gains tend to show diminishing returns as LR approaches 1.0. Since KIR is 1.0 in this setup, F1-open is consistently 0, as no classes were designated as unknown.

Table 5.8: Performance of ADB on our labeled dataset on varying degrees of labeled data, while keeping the known class ratio fixed.

Dataset	F1-known	F1-open	F1	Acc	Method	Backbone	known_intent_ratio	labeled_ratio	LossFn	Seed	Epochs
labeled_dataset_stratified	69.8816	0.0000	68.6110	79.10	ADB	bert	1.00	0.3	CrossEntropyLoss	0	100
labeled_dataset_stratified	77.7429	0.0000	76.3294	80.60	ADB	bert	1.00	0.4	CrossEntropyLoss	0	100
labeled_dataset_stratified	78.5840	0.0000	77.1552	82.09	ADB	bert	1.00	0.5	CrossEntropyLoss	0	100
labeled_dataset_stratified	83.5058	0.0000	81.9876	84.48	ADB	bert	1.00	0.6	CrossEntropyLoss	0	100
labeled_dataset_stratified	80.2103	0.0000	78.7519	81.79	ADB	bert	1.00	0.7	CrossEntropyLoss	0	100
labeled_dataset_stratified	84.4547	0.0000	82.9192	83.28	ADB	bert	1.00	0.8	CrossEntropyLoss	0	100
labeled_dataset_stratified	85.6676	0.0000	84.1100	84.48	ADB	bert	1.00	0.9	CrossEntropyLoss	0	100
labeled_dataset_stratified	85.7999	0.0000	84.2399	84.78	ADB	bert	1.00	1.0	CrossEntropyLoss	0	100

IMPACT OF KNOWN INTENT RATIO (KIR). Table 5.9 investiges the algorithm's performance under the more challenging open-set scenario by varying the KIR at 0.25, 0.50, and 0.75, while keeping LR fixed at 1.0 (using all available labeled data). This setup compares both versions of our dataset (`labeled_dataset_stratified` and the original `labeled_dataset` splits) against standard benchmarks.

Across all datasets, a general trend is observed where F1-known tends to increase with higher KIR, as the model benefits from seeing more classes during training for the closed-set part of the task. Conversely, the F1-open score, measuring the ability to detect unknown intents, shows more complex behavior.

Table 5.9: Performance of ADB for varying degrees of known intent ratios, while keeping the labeled ratio fixed.

Dataset	F1-known	F1-open	F1	Acc	Method	Backbone	known_intent_ratio	labeled_ratio	LossFn	Seed	Epochs
labeled_dataset_stratified	55.6016	89.0511	57.8315	81.94	ADB	bert	0.25	1.0	CrossEntropyLoss	0	100
labeled_dataset_stratified	73.0727	77.7328	73.2392	74.63	ADB	bert	0.50	1.0	CrossEntropyLoss	0	100
labeled_dataset_stratified	83.1964	76.0736	83.0227	80.90	ADB	bert	0.75	1.0	CrossEntropyLoss	0	100
labeled_dataset	70.582	93.908	72.137	90.12	ADB	bert	0.25	1.0	CrossEntropyLoss	0	100
labeled_dataset	78.123	85.288	78.343	83.74	ADB	bert	0.50	1.0	CrossEntropyLoss	0	100
labeled_dataset	83.140	72.283	82.875	83.74	ADB	bert	0.75	1.0	CrossEntropyLoss	0	100
BANKING77	72.909	85.117	73.520	79.35	ADB	bert	0.25	1.0	CrossEntropyLoss	0	100
BANKING77	80.549	78.019	80.484	78.28	ADB	bert	0.50	1.0	CrossEntropyLoss	0	100
BANKING77	87.301	71.889	87.040	82.95	ADB	bert	0.75	1.0	CrossEntropyLoss	0	100
CLINC150	80.647	90.840	80.910	87.11	ADB	bert	0.25	1.0	CrossEntropyLoss	0	100
CLINC150	88.824	85.949	88.786	86.80	ADB	bert	0.50	1.0	CrossEntropyLoss	0	100
CLINC150	92.648	78.009	92.519	88.71	ADB	bert	0.75	1.0	CrossEntropyLoss	0	100
StackOverflow	78.349	90.795	80.423	86.60	ADB	bert	0.25	1.0	CrossEntropyLoss	0	100
StackOverflow	87.275	89.293	87.459	88.35	ADB	bert	0.50	1.0	CrossEntropyLoss	0	100
StackOverflow	87.952	75.700	87.187	85.17	ADB	bert	0.75	1.0	CrossEntropyLoss	0	100

- **Performance on Custom Dataset Splits**: Comparing the `labeled_dataset_stratified` and original `labeled_dataset` (non-stratified) ver-

sions reveals performance variations depending on the KIR and metric. For instance, at KIR=0.75, the stratified version achieves rather similar F1-known (83.2% vs. 83.14%) compared to its non-stratified counterpart, however for the F1-open, the stratified version reaches better results (76.07% vs. 72.28%). At lower KIR values (0.25, 0.50), the original splits sometimes showed higher F1-known and F1-open scores. This could suggest the stratification to lead to more stable OOD detection, however as only one random seed was used to produce results, more experiments should be conducted to find whether the produced metrics occur systematically.

- **Comparison with Benchmarks**: When compared to established benchmarks under the same KIR conditions, our corporate email intent dataset proves challenging, particularly for OOD detection. While the F1-known scores on our dataset (both versions) are often comparable or higher than those on BANKING77 and StackOverflow, and sometimes approach CLINC150 levels, the F1-open scores are consistently lower than those achieved on BANKING77 and CLINC150 across all tested KIR values. StackOverflow generally also has low F1-open scores, sometimes similar to our dataset. This indicates that the ADB model struggles more significantly to distinguish unknown utterances from the known, fine-grained intents within our dataset compared to benchmarks with greater topical diversity (CLINC150) or perhaps stronger keyword signals (StackOverflow).

These findings align with the dataset characteristics identified earlier. The difficulty in OOD detection (low F1-open) suggests that the semantic closeness of the fine-grained intents makes it hard for ADB to establish clear, encompassing decision boundaries that effectively exclude unseen concepts.

5.4.2 *Open Intent Discovery (Deep Aligned Performance)*

This section examines the performance of the Deep Aligned Clustering algorithm [Zha+21b], a representative method for semi-supervised intent discovery, on our labeled corporate email intent dataset. The experiments, run using the TEXTOIR framework, evaluated the algorithm's ability to cluster utterances and discovery underlying intents under varying KIRs while operating in a low-resource setting with the LR fixed at 10%. The primary evaluation metrics reported are Accuracy, ARI, and NMI. Table 5.10 presents these findings.

It is important to note that the results for our dataset versions (`labeled_dataset_stratified` and `labeled_dataset`) were obtained from runs using a single random seed (Seed 0). The benchmark dataset results included in the table are referenced from the original Deep Aligned Clustering paper [Zha+21b] for context, as obtaining these was not feasible due to time constraints.

Examining the performance on our custom datasets reveals the following trends:

- **Impact of Known Intent Ratio (KIR):** For both the stratified and non-stratified versions of our dataset, there is a general trend of improvement across all reported metrics (Accuracy, ARI, NMI) as the KIR increases from 0.25 to 0.75. For instance, on the non-stratified `labeled_dataset`, Accuracy increases from 34.50% at KIR=0.25 to 52.75% at KIR=0.75, ARI increases from 24.64 to 44.18, and NMI increases from 62.74 to 76.04. This indicates that providing the model with knowledge of more intent classes during training assists the Deep Aligned Clustering algorithm in producing better overall clustering structures, even with only 10% labeled data.

- **Comparison of Custom Dataset Splits:** When comparing the performance between the `labeled_dataset_stratified` and the original `labeled_dataset` (non-stratified) versions under these low-LR conditions, the non-stratified version consistently achieved higher scores across all metrics for the tested KIR values. For example, at KIR=0.75, the non-stratified dataset reached an Accuracy of 52.75%, ARI of 44.18, and NMI of 76.04, compared to 43.28%, 22.94, and 70.49, respectively, for the stratified version. While these single-seed results suggest the original data split might be more conducive to Deep Aligned Clustering in this specific low-label setting, further experiments with multiple seeds would be needed to confirm the robustness of this observation.

- **Comparison with Benchmark Datasets:** When comparing the performance on our corporate email dataset (both versions) to the benchmark datasets under these identical low-label (LR=0.1) and varying KIR conditions, a clear performance gap is evident. Our dataset consistently yields lower scores across Accuracy, ARI, and NMI compared to BANKING77 and CLINC150 at all tested KIR levels (0.25, 0.50, 0.75). CLINC150 demonstrates the strongest performance overall, achieving substantially higher ARI and NMI scores, suggesting its structure is more amenable to discovery by Deep Aligned Clustering even with limited labeled data. Performance

on our dataset is also generally lower than on StackOverflow, particularly in terms of ARI, although NMI scores are sometimes comparable or slightly higher at lower KIR values compared to StackOverflow. For example, at KIR=0.75, the highest ARI achieved on our dataset (44.18 on the non-stratified version) is considerably lower than that on BANKING77 (53.09), StackOverflow (60.09), and CLINC150 (79.94). This pattern holds across metrics and KIR levels, positioning our dataset as demonstrably more challenging for the Deep Aligned Clustering algorithm under these specific semi-supervised, low-resource conditions compared to these standard benchmarks.

Overall, the results indicate that Deep Aligned Clustering can leverage partial supervision (10% LR) to discover intent structures in the corporate email dataset, with performance improving as more classes are known during training. However, the clustering quality, particularly reflected by the ARI scores, remains moderate, highlighting the difficulty of this task on the dataset.

Table 5.10: Performance of Deep Aligned for varying degrees of known intent ratios, while keeping the labeled ratio fixed (at 10%). Due to time constraints the metrics reported for benchmark datasets have been taken from the original Deep Aligned paper [Zha+21b].

Dataset	Accuracy	ARI	NMI	Method	Backbone	known_intent_ratio	labeled_ratio	Seed	Epochs
labeled_dataset_stratified	34.03	20.54	61.56	DeepAligned	bert	0.25	0.1	0	100
labeled_dataset_stratified	37.01	20.83	65.64	DeepAligned	bert	0.50	0.1	0	100
labeled_dataset_stratified	43.28	22.94	70.49	DeepAlignedB	bert	0.75	0.1	0	100
labeled_dataset	34.50	24.64	62.74	DeepAligned	bert	0.25	0.1	0	100
labeled_dataset	42.25	30.68	67.17	DeepAligned	bert	0.50	0.1	0	100
labeled_dataset	52.75	44.18	76.04	DeepAlignedB	bert	0.75	0.1	0	100
BANKING77	49.08	37.62	70.50	DeepAligned	bert	0.25	0.1	n/a	100
BANKING77	59.38	47.95	76.67	DeepAligned	bert	0.50	0.1	n/a	100
BANKING77	64.63	53.09	79.39	DeepAligned	bert	0.75	0.1	n/a	100
CLINC150	74.07	64.63	88.97	DeepAligned	bert	0.25	0.1	n/a	100
CLINC150	80.70	72.56	91.59	DeepAligned	bert	0.50	0.1	n/a	100
CLINC150	86.79	79.94	93.92	DeepAligned	bert	0.75	0.1	n/a	100
StackOverflow	54.50	37.96	50.86	DeepAligned	bert	0.25	0.1	n/a	100
StackOverflow	74.52	57.62	68.28	DeepAligned	bert	0.50	0.1	n/a	100
StackOverflow	77.97	60.09	73.28	DeepAligned	bert	0.75	0.1	n/a	100

5.5 LABEL VERIFICATION USING CLEANLAB

To assess the consistency of the labels generated through the semi-automated workflow and to identify potential areas of ambiguity or conflict within the dataset, a label quality assessment was conducted using the Cleanlab framework. This analysis utilized the out-of-sample predicted probabilities generated via 5-fold cross-validation, as detailed in Section 4.2.3.

The Cleanlab analysis flagged a total of 281 utterances, representing approximately **4.2%** of the 6,691 unique labeled data points (excluding the test set's OOD category for this specify analysis), as potential label inconsistencies based on the model's predictions versus the assigned labels.

Visualizations help illustrate the patterns within these flagged instances. The label conflict heatmap (Figure 5.7) highlights the specific pairs of assigned labels and model-suggested labels that occurred most frequently among the 281 flagged items. Notably, significant counts appear for confusion between semantically close intents, such as `request_meeting` and `propose_meeting`. The network graph (Figure 5.8) further visualizes these relationships, showing that classes like `request_meeting`, `request_availability`, `offer_assistance` are among the central nodes with numerous incoming and outgoing conflict edges, indicating they are frequently involved in potential inconsistencies.

Qualitative examination of the top-ranked potential label issues identified by Cleanlab reveals important characteristics about the nature of these flagged inconsistencies. **Crucially, inspection suggests that many of these flagged instances do not necessarily represent simple annotation errors, but rather highlight inherent complexities of the dataset and the labeling task itself.** Frequently observed patterns in the flagged data include:

- **Multi-Intent Utterances:** Many flagged sentences appear to convey multiple, distinct communicative simultaneously according to the established taxonomy. For example, utterances requesting both sending a document and being copied (`request_send_document` vs. `request_add_cc`), or expressing urgency while requesting submission (`request_urgency_with_deadline` vs. `request_send_document`), were flagged. In such cases, the Cleanlab-suggested label often highlights a different, yet potentially equally valid, facet of the utterance's meaning compared to the originally assigned single label.
- **Semantic Overlap and Fuzzy Boundaries:** The high confusion rates between classes like `request_meeting` and `propose_meeting`, or `offer_assistance` and `request_call`, evident in the heatmap and network graph, are reflected in the flagged examples. These often represent utterances lying in ambiguous semantic territory where subtle phrasing differences or missing context make a definitive single-label assignment challenging, leading to disagreement between the assigned label and the model's prediction based on learned patterns.

In summary, the Cleanlab analysis quantified potential label inconsistencies at approximately 4.2%. More significantly, the qualitative inspection of these flagged instances

strongly suggests that many reflect fundamental challenges inherent in applying a single-label classification scheme to fine-grained intents within conversational corporate email, even when attempting to simplify the problem by focusing on sentence-level analysis, particularly concerning the prevalence of multi-intent expressions and the fuzzy boundaries between closely related semantic categories.

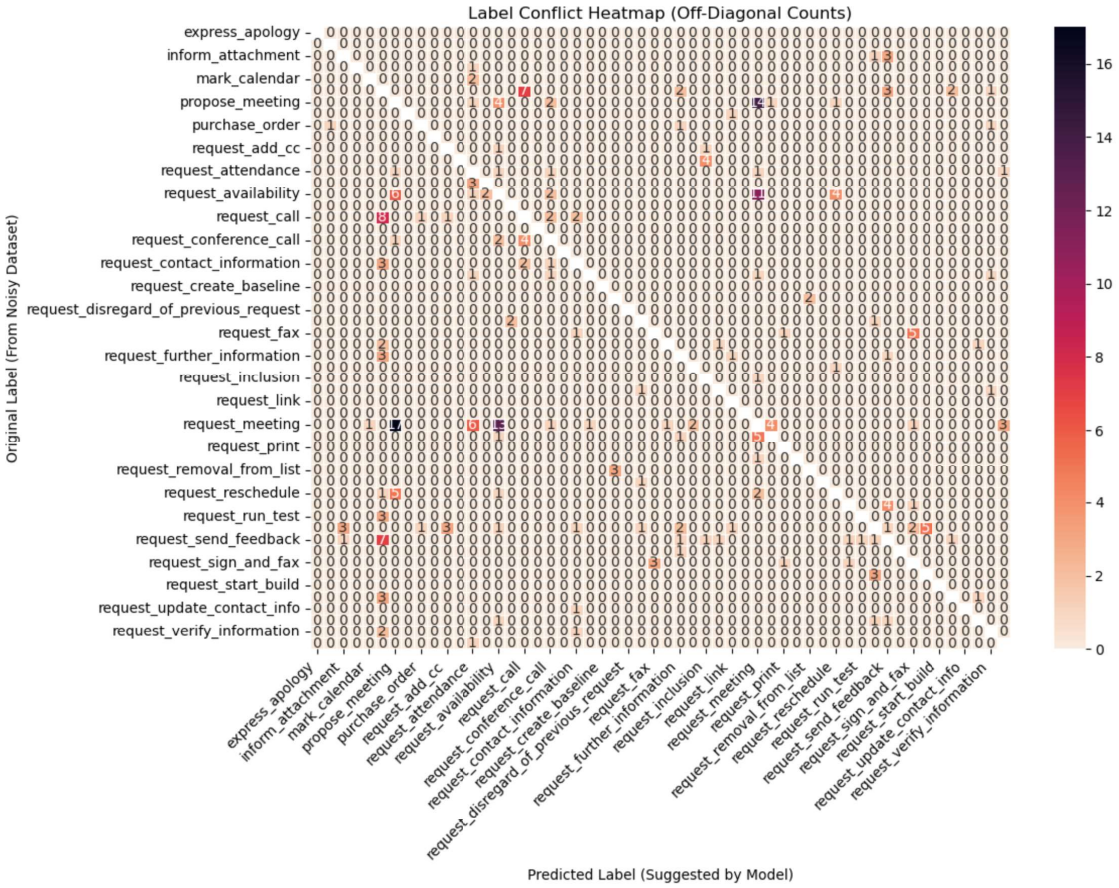

Figure 5.7: Matrix of the label conflicts obtained through Cleanlab. It shows every conflict for our assigned labels with a label assigned by a SetFit classifier that has been trained using a 5-fold cross-validation.

Figure 5.8: Network graph illustrating label conflicts detected by Cleanlab. Nodes are classes, sized by total off-diagonal conflict involvement. Edges depict specific conflicts (Original Label -> Predicted Label), with thickness proportional to the conflict count. The layout reveals structural relationships between confused classes.

6

DISCUSSION

This chapter interprets the findings presented in Chapter 5, contextualizing the characteristics of the generated labeled corporate email intent dataset and evaluating the effectiveness and outcomes of the developed workflow. The discussion addresses the research questions outlined in Chapter 1, reflects on the challenges inherent in analyzing corporate email intents and acknowledges the limitations of this study.

6.1 SUMMARY OF FINDINGS

The research implemented a semi-automated workflow to produce a labeled dataset of 6,785 corporate email utterances across 54 intent classes. Key results indicated:

- A dataset composition heavily skewed towards common operational intents.
- Significant semantic overlap between many fine-grained intent classes (low Silhouette Score: 0.057, high DBI: 3.257).
- High intra-class similarity for some intents but also high inter-class similarity between related intents.
- Lexical analysis confirmed shared vocabulary across many classes.
- Structural comparisons positioned the dataset closer to BANKING77 in complexity due to semantic similarity, differing from CLINC150's topical diversity or StackOverflow's keyword reliance.
- Label quality assessment via Cleanlab flagged 4.2% of labels as potential issues, often **highlighting instances reflecting semantic overlap, fuzzy category boundaries, or multi-intent utterances inconsistent with the single-label assignment**.
- Supplementary random sampling confirmed the sparsity of most defined intents in the original corpus.

6.2 INTERPRETATION OF FINDINGS IN RELATION TO RESEARCH QUESTIONS

This section synthesizes the key findings presented in Chapter 5 and interprets them directly in the context of the research questions posed in Section 1.2. By examining the effectiveness and challenges of the proposed methodology (RQ1), the structural characteristics of the resulting dataset compared to benchmarks (RQ2), and the insights gained from label quality assessment (RQ3), we can draw conclusions about the complexities of fine-grained intent discovery in corporate email and the contributions of this work.

6.2.1 *RQ1: Identifying, Categorizing, and Labeling Intents with LLM Facilitation*

The first research question explored methods for effectively identifying, categorizing and labeling fine-grained intents in corporate emails, considering the potential role of LLMs. The developed workflow (Chapter 3) demonstrated a viable semi-automated approach, offering a more scalable alternative to purely manual annotation for large corpora. Key insights regarding this question include:

WORKFLOW VIABILITY & LLM FACILITATION. The workflow, using LLMs for quality scoring and rich feature generation (purpose summaries), successfully enabled clustering and dataset creation where simpler methods might have failed due to semantic nuance. LLMs acted as crucial **facilitators and weak supervisors**, bootstrapping the process. However, the need for prompt refinement and the residual inconsistencies suggest LLMs require careful integration and human oversight.

CHALLENGES IN CATEGORIZATION & LABELING. The process highlighted inherent difficulties. While the workflow surfaced potential categories, the subsequent qualitative analysis and Cleanlab results revealed that **drawing clear, distinct lines between conceptually close intents is challenging**. The very act of assigning a single label, even at the sentence level, often proved problematic, as discussed further under RQ3. The workflow surfaced these challenges but didn't automatically resolve the inherent fuzziness, indicating taxonomy definition and refinement remain critical, expert-driven tasks.

EFFECTIVENESS OF THE ALTERNATING REFINEMENT STRATEGY. The iterative workflow employed an alternating strategy: using classifiers to populate known classes and re-clustering the remainder to discover potential new ones. The classifier-based population phase effectively increased example counts for established classes. The re-clustering phase, applied to the progressively smaller unlabeled pool, did surface additional clusters. However, manual inspection revealed challenges: Some clusters grouped utterances based on overly specific LLM annotations (occasionally including entities like names), fragmenting semantically similar intents rather than revealing distinct new categories. This required careful manual merging or re-evaluation. Other clusters contained utterances genuinely hard to categorize within the existing fine-grained taxonomy. These often involved very specific, low-frequency requests or suffered significantly from the lack of conversational context, making a definitive single-label assignment problematic even for a human reviewer. Therefore, while the alternating process allowed for exploration, it does not come without its limitations, highlighted by the difficulty of categorizing niche or context-dependent utterances and the potential for LLM annotations to sometimes be overly specific, demanding significant human effort in validation and taxonomy management rather than straightforward discovery of wellformed new intents.

6.2.2 *RQ2: Structural Characteristics and Benchmark Comparison*

The second research question investigated the structural properties of the generated dataset and its relation to established benchmarks. The analyses presented in Chapter 5 reveal a dataset primarily characterized by a highly skewed intent distribution and significant semantic overlap. While the skew towards common operational intents is partly an artifact of the targeted discovery methodology, supplementary random sampling confirmed the inherent sparsity of most fine-grained intents within the original corpus. Beyond distribution, the dataset is marked by considerable semantic overlap among its 54 fine-grained classes. This was consistently observed through quantitative metrics, such as the low Silhouette Score (0.057) and high DBI (3.257), as well as qualitative UMAP visualizations depicting fluid cluster boundaries. Although some classes exhibit reasonable internal cohesion, the relatively high average inter-class similarity (0.172) and supporting lexical analysis confirm that many distinct intents are expressed using shared vocabulary, posing a significant challenge for discrimination.

The inherent difficulties suggested by these structural characteristics were empirically validated through experiments using the TEXTOIR framework (Section 5.4). When evaluating the ADB algorithm for open intent detection, the dataset proved challenging, particularly for identifying unknown (OOD) intents. While performance on classifying *known* intents was often comparable to or better than benchmarks like BANKING77 and StackOverflow under similar conditions Table 5.9, the F1-score for OOD detection was consistently lower than that achieved on BANKING77 and CLINC150. This suggests that the high semantic overlap and fuzzy boundaries within our dataset make it difficult for ADB to establish clear decision perimeters that effectively exclude closely related but unseen concepts, a task potentially aided by greater topical diversity in datasets like CLINC150.

This challenge posed by semantic overlap was further underscored by the intent discovery results using Deep Aligned Clustering (Table 5.10). Operating under identical low-label (LR=0.1) and varying KIR conditions, our dataset consistently yielded lower Accuracy, ARI, and NMI scores compared to all benchmark datasets. The particularly low ARI scores highlight significant difficulty in forming clusters that align well with the fine-grained ground-truth labels, even with partial supervision. This indicates that the inherent blending of intents within the embedding space makes their unsupervised or semi-supervised discovery substantially more difficult in this corporate email context compared to the standard benchmark domains.

Contextualizing these findings, the dataset's combination of reasonable intra-class cohesion coupled with high inter-class semantic similarity most closely mirrors the structural challenges of BANKING77, suggesting difficulty arises from needing to disambiguate functionally related intents using overlapping language. This contrasts with the topical diversity of CLINC150 or the keyword reliance of StackOverflow. Therefore, the combination of fine granularity, high semantic overlap, lack of strong unique keywords, underlying sparsity, and the empirical performance results positions this dataset as a demanding benchmark for models requiring nuanced semantic understanding, especially for open-set recognition and low-resource discovery. However, it remains

crucial to acknowledge that the 54-class taxonomy is not exhaustive, representing only the intents discovered and populated via the applied methodology; strong performance here reflects robustness to ambiguity within this scope, not necessarily comprehensive open-world email intent classification proficiency.

6.2.3 *RQ3: Assessing Label Quality and Domain Challenges.*

The third research question addressed the assessment of label quality and the fundamental domain challenges revealed thereby. The application of Cleanlab provided a systematic method for probing label consistency, identifying approximately 4.2% of labels as potential issues based on OOS model predictions. More importantly than the quantity, the qualitative examination of these flagged instances (Section 5.5) revealed their nature: they frequently stemmed not from simple annotation errors, but from the inherent complexities of the domain. Key patterns included the prevalence of multi-intent utterances, where a single sentence serves multiple communicative goals (e.g., requesting a document *and* requesting to be CC'd), and utterances falling on inherently fuzzy semantic boundaries between closely related categories (e.g., `request_meeting` vs. `propose_meeting`), where subtle phrasing or missing context makes definitive single-label assignment difficult even for humans.

Crucially, the nature of these label conflicts diagnosed by Cleanlab provides a direct explanation for the performance limitations observed in the TEXTOIR experiments. The prevalence of multi-intent and semantically ambiguous utterances inherently complicates the task for single-label algorithms like ADB. Such instances are logically difficult for the model to classify correctly against a single ground-truth label or to confidently reject as OOD when they share features with known classes, thus contributing to the lower F1-open scores observed. Similarly, for intent discovery, if utterances naturally bridge multiple ground-truth categories, as suggested by the Cleanlab conflicts and high inter-class similarity metrics, clustering algorithms like Deep Aligned will inevitably struggle to partition them into pure, distinct groups matching the predefined taxonomy, resulting in the lower ARI and NMI scores seen in the experiments.

Therefore, the Cleanlab assessment functions as more than just a noise estimation tool; it serves as a diagnostic instrument confirming the fundamental challenges of labeling fine-grained intents in this domain. The identified inconsistencies largely reflect the prevalence of multi-intent expressions and semantic ambiguity inherent in conversational corporate email, which clash with the constraints of the single-label, sentence-level classification paradigm employed. These inherent domain complexities, rather than simple annotation mistakes, are significant contributors to the difficulties faced by standard intent detection and discovery algorithms, as demonstrated empirically by the TEXTOIR results. This underscores the limitations of the chosen simplified approach and points towards the need for methods better suited to the multi-faceted nature of email communication.

6.3 IMPLICATIONS OF THE FINDINGS

The combined results and interpretations carry several implications for the field of NLP, particularly concerning email analysis and intent understanding.

THE DIFFICULTY OF REAL-WORLD EMAIL INTENT. This study underscores that extracting fine-grained intents from diverse corporate email is substantially more challenging than often assumed based on performance on standard, often cleaner, benchmark datasets. The high degree of semantic overlap, reliance on context, conversational nature, and prevalence of multi-intent utterances necessitate models and approaches that go beyond simple classification of isolated sentences. Progress in practical email automation (e.g., summarization, task extraction) will likely depend on effectively tackling these complexities.

LLMS AS POWERFUL BUT IMPERFECT TOOLS FOR DISCOVERY. LLMs demonstrably facilitate the discovery process by generating richer semantic features than previously feasible at scale, enabling clustering based on nuanced meaning. However, they are not a silver bullet. Their outputs can be ambiguous, overly specific, or fail to capture implicit meaning without careful prompt engineering and human validation. Their most effective role in such workflows appears to be as sophisticated weak supervision providers or feature generators within a human-in-the-loop system, rather than fully autonomous labelers.

LIMITATIONS OF SINGLE-LABEL SENTENCE-LEVEL ANALYSIS. The consistent emergence of multi-intent examples and context-dependent ambiguities, highlighted particularly by the Cleanlab analysis, strongly suggests the limitations of the prevalent single-label, sentence-level classification paradigm for accurately representing email communication. Future work likely needs to embrace multi-label frameworks and incorporate broader context (email body, thread history) for higher fidelity.

VALUE OF REALISTIC DATASETS. The generated dataset, precisely *because* it reflects the skew, sparsity, overlap, and labeling ambiguities encountered through a practical discovery process on real-world data, serves as a valuable and challenging benchmark. Evaluating models on such datasets may provide a more realistic assessment of their capabilities for handling genuine corporate communication compared to performance on more sanitized or topically diverse benchmarks.

6.4 LIMITATIONS OF THE STUDY

It is essential to acknowledge the limitations inherent in this research, which frame the scope and generalizability of the findings:

SINGLE ANNOTATOR AND LABEL ROBUSTNESS. The entire manual labeling process, including the initial inspection of clusters, refinement, and the supplementary random

sampling analysis, was conducted by a **single annotator (the author)**. While efforts were made towards consistency, this inherently limits the robustness of the assigned labels. **Utilizing multiple annotators and measuring Inter-Annotator Agreement (IAA)** would be crucial for developing a truly production-grade, reliable dataset. Disagreements between annotators could have further highlighted inherently ambiguous cases or weaknesses in the taxonomy definition. Consequently, the current dataset and workflow leading to it should be viewed more as a **proof-of-concept demonstrating the methodology and highlighting domain challenges**, rather than a fully validated, gold-standard labeling approach ready for direct deployment without further verification.

SINGLE CORPUS. The preliminary reliance on the Avocado dataset means that the specific intent taxonomy, observed distributions, and potentially the degree of semantic overlap might reflect the communication patterns unique to that specific (now-defunct) IT company. Generalizability to other corporate environments requires further investigation.

LLM DEPENDENCE. The results related to LLM annotation and feature generation are specific to LLaMA 3 8B Instruct and the prompt strategies employed. Different models (e.g., larger models, different architectures) or alternative prompting techniques might yield quantitatively or qualitatively different outcomes.

SINGLE-LABEL METHODOLOGICAL CONSTRAINT. The workflow fundamentally operated under a single-label assumption per sentence. This simplification does not fully capture the multi-intent nature of many email utterances, and this limitation is directly reflected in the label quality assessment results.

SENTENCE-LEVEL FOCUS. By design, the analysis focused on isolated sentences, deliberately excluding email-level or thread-level context, which is often crucial for disambiguation intent in real conversations.

EVALUATION METRICS. The quantitative metrics used (Silhouette, DBI, cosine similarity) have known limitations regarding high-dimensional spaces, providing only partial insights into the complex semantic structure.

CLEANLAB ASSESSMENT. The accuracy of the label quality assessment relies on the quality of the OOS probabilities generated by the cross-validated SetFit/Logistic Regression models. Errors or biases in these probabilities could affect the identification of label issues.

VALUE AND LIMITATIONS OF THE GENERATED DATASET. The generated dataset, *because* it encapsulates the observed skew, sparsity, semantic overlap, and labeling ambiguities encountered through a practical discovery process, serves as a valuable resource. It provides a **realistic testbed** for developing and evaluating models robust to the specific challenges inherent in fine-grained corporate email intent analysis. However, its

non-exhaustive nature must be acknowledged. High performance achieved by models trained solely on this dataset reflects an ability to navigate ambiguity *within the defined 54-class taxonomy*, but **does not necessarily equate to general proficiency** in classifying the potentially vast and open ended set of intents present in unrestricted corporate communications. It serves best as a benchmark for specific robustness characteristics rather than overall open-domain email understanding.

7

CONCLUSION AND FUTURE WORK

This final chapter synthesizes the research effort undertaken in this thesis. It begins by summarizing the core problem, objectives, and key findings derived from the investigation into fine-grained intent discovery and labeling within corporate email. Subsequently, it highlights the principal contributions of this work to the field. Finally, based on the insights gained and the limitations acknowledged throughout the study, it proposes several promising directions for future research.

7.1 CONCLUSION

This thesis addressed the significant challenge of discovering and labeling fine-grained user intents within the complex and often ambiguous domain of corporate email communication. Motivated by the lack of suitable datasets and the limitations of applying standard NLP techniques directly, the primary objective was to develop and evaluate a systematic workflow for this task, characterize the resulting dataset, and assess the quality of the generated labels.

To achieve this, a semi-automated, multi-stage workflow was designed and implemented (RQ1), leveraging initial rule-based filtering, LLM-based quality scoring and feature generation (using LLaMA 3 8B Instruct), semantic clustering, and an iterative refinement process involving classifiers like SetFit. This workflow proved capable of processing a large unlabeled corpus (the Avocado dataset) and generating a substantial labeled dataset comprising 6,785 utterances across 54 distinct, fine-grained intent classes reflecting typical corporate operations.

The structural analysis of this dataset (RQ2) revealed characteristics distinct from standard benchmarks. It exhibits significant intent frequency skew, high semantic overlap between related classes (low Silhouette Score: 0.057, high DBI: 3.257), and relies heavily on shared vocabulary, positioning its complexity profile closest to challenging benchmarks like BANKING77. Crucially, supplementary analysis indicated that most identified intents are likely sparse in the original corpus, highlighting that the final dataset, while realistic in its challenges, is not exhaustive nor representative of raw frequencies.

The assessment of label quality using Cleanlab (RQ3) identified potential inconsistencies in approximately 4.2% of the labels. More importantly, the nature of these flagged issues underscored fundamental domain challenges: the prevalence of multi-intent utterances and the inherent fuzziness of boundaries between subtle, fine-grained intents often clash with the single-label, sentence-level approach employed. This suggests that many inconsistencies reflect the complexity of the communication itself rather than simple annotation errors, a finding reinforced by the single-annotator nature of this study's labeling process.

In conclusion, this research contributes: (1) A novel, adaptable workflow demonstrating the utility (and limitations) of LLMs as weak supervisors for intent discovery in a complex domain. (2) A new, characterized labeled dataset reflecting the real-world challenges of fine-grained intent ambiguity and overlap in corporate email. (3) An analysis highlighting the structural properties of such data compared to benchmarks and diagnosing label quality issues tied to inherent domain characteristics. While demonstrating a viable path forward, the findings emphasize the need for context-aware, potentially multi-label approaches to fully capture the richness of email communication.

7.2 FUTURE WORK

Based on the experiences and findings of this research, particularly the challenges surrounding ambiguity and multi-intent utterances, future work should focus on evolving the methodology and modeling approaches in the following key directions:

- **Adopting a Flexible Multi-Label Framework:** The current single-label assignment often forced a choice when utterances contained multiple communicative goals, a common occurrence highlighted by the LLM-generated features and the Cleanlab analysis. Future work should transition to a **true multi-label annotation framework**. This means allowing annotators (ideally multiple, for consensus) to assign **any combination of relevant atomic intent labels** (from the established or refined taxonomy) to a single utterance. For instance, an email sentence requesting a document *and* asking for it to be faxed could be labeled with both request_send_document *and* request_fax. This approach offers greater fidelity to the complex nature of email communication and directly addresses the limitations encountered.

- **Exploring Hierarchical Intent Structures:** To better organize the potentially large set of fine-grained intents and aid both annotation consistency and modeling, future research could investigate structuring the intent taxonomy **hierarchically**. This might involve defining broader, coarse-grained categories (e.g., `Information Exchange`, `Task Management`, `Scheduling`, `Social Communication` as suggested by prior work, or data-driven categories) under which the fine-grained intents reside. For example, `request_meeting` and `request_reschedule` could fall under "Scheduling". This structure could:
  - Guide annotators by providing layered choices.
  - Enable models to potentially predict coarse categories first, then refine to specific fine-grained intents (or multiple intents within/across branches).
  - Offer different levels of granularity for downstream applications.

  This hierarchical view complements the multi-label approach by providing organizational structure to the set of potentially co-occurring fine-grained labels.

- **Leveraging Context for Enhanced Feature Generation:** The current sentence-level analysis inherently misses contextual cues that could significantly aid intent

disambiguation. A natural direction for future work is to explore methods for incorporating broader context (e.g., the full email body, subject line, or preceding messages) into the LLM feature generation process. The objective would be to provide the LLM with richer information to produce more accurate semantic features (like purpose summaries or implicit/explicit intent descriptors) for the target sentence. However, achieving this effectively presents a known challenge: ensuring the LLM utilizes the provided context appropriately for interpreting the specific target sentence, rather than simply extracting information from the context itself. Developing robust prompting strategies or employing LLMs with advanced capabilities in focused, context-aware reasoning will be key to successfully implementing this enhancement and improving the quality of the initial intent signals.

Pursuing these directions—specifically embracing flexible multi-labeling, potentially organizing intents hierarchically, and refining context utilization—will be crucial for developing NLP systems that can more accurately capture and act upon the nuanced communicative intents prevalent in real-world corporate email.

Part II

APPENDIX

A

TABLES

Table A.1: Comparison Table of Intent Detection approaches found in the literature

Approach / Model Name	Core Idea / Method	How Unknowns Handled	Underlying Tech / Components	Motivation / Addressed Limitation
Traditional Classification (Closed-World)	Map utterance to fixed, known classes.	Assumes closed world; does not handle unknown intents.	General classification models.	Baseline approach; simple setup.
OpenMax [BB16]	Replace Softmax with distance-aware layer; measure distance to known class mean activations.	Classify as unknown if input activation is far from all known class means.	DNN activations, Distance calculation.	Softmax limitation in OSR; provides a mechanism to reject unknowns.
1-vs-Rest Output [SXL17]	Replace Softmax with independent sigmoid activation per class.	Allows low confidence scores across all known classes, enabling rejection of inputs not fitting any category.	Sigmoid activation functions.	Softmax forces assignment even for unrelated samples; 1-vs-rest allows non-assignment.
BiLSTM + LMCL + LOF [LX19]	Two-stage: 1) BiLSTM feature extraction + LMCL loss for compact and separated features. 2) LOF density check on features.	Unknowns identified as low-density outliers in the feature space by LOF.	BiLSTM, LMCL, LOF.	Traditional Softmax doesn't enforce separation needed for density methods; LMCL creates better feature space for LOF.
SEG [Fan+20]	Model embeddings with GMM, inject class semantics. Use density-based outlier detection (e.g., LOF).	Density-based outlier detection is more effective on the dense, ball-like clusters formed by GMM.	GMM, Class Semantics, Density Outlier Detection (LOF).	Addresses LMCL limitations (elongated clusters unsuitable for LOF, ignores label semantics).
ADB [ZXL21]	Use BERT features; learn adaptive spherical decision boundary per known class. Balance empirical vs. open space risk.	Explicitly defines boundaries enclosing known intents; unknowns fall outside. Boundaries adjust dynamically.	BERT, Spherical boundary learning, Risk balancing.	Addresses GMM/LOF limitations (needs architecture changes, no explicit boundaries); dynamic adjustment.

Table A.2: Comparison Table of Intent Discovery approaches found in the literature

Approach / Model Name	Setting	Core Idea / Method	Key Innovation / Feature	How Labeled Data Used (if Semi-Sup)	Motivation / Addressed Limitation
Traditional Clustering (K-Means, Agglom.) [Mac+67], [GK78]	Unsup.	Group utterances based on feature similarity/ or distance.	Baseline clustering algorithms.	N/A	Simple grouping, but struggles with high-dim data.
Deep Clustering (General) [XGF16], [Yan+17]	Unsup.	Simultaneously learn feature representations (via DNNs) and cluster assignments.	Joint optimization of feature learning and clustering.	N/A	Improves over sequential approach (feature extraction then clustering).
CC [Li+21]	Unsup.	Maximize similarity of augmented positive pairs, minimize for negative pairs (instance and cluster levels).	Dual contrastive learning framework.	N/A	Enhances cluster separation (initially noted in CV).
SCCL [Zha+21a]	Unsup.	Apply CC principles to short text using specific text augmentations (WordNet, Contextual, Back-translation).	Adapting CC effectively for text data augmentations.	N/A	Builds on CC success, applies it to NLP task.
CDAC+ [LXZ19]	Semi-Sup.	Frame as pairwise classification (same/ or diff intent) using BERT similarity and thresholding; refine via self-training.	Pairwise constraints from labels; dynamic thresholding; self-training.	Small set of labeled intent pairs for pairwise constraints.	Criticizes intensive feature engineering; aims for robustness with limited supervision.
Deep Aligned Clustering [Zha+21b]	Semi-Sup.	Pre-train on labeled data; cluster features; use k-means pseudo-labels aligned across epochs via centroid alignment (Hungarian algo).	Pre-training step; Centroid alignment strategy.	Used for supervised pre-training loss.	Addresses CDAC+'s weak supervision issues with mixed intents and poor knowledge transfer; stabilizes clustering.
Multi-Task Pre-train + Neighborhood Contrastive [Zha+22a]	Semi-Sup.	Pre-train LM on public + target data; use neighborhood-aware contrastive loss objective incorporating semantics.	Multi-task pre-training; Neighborhood-aware contrastive loss.	Labeled target data used alongside public and unlabeled data during pre-training.	Aims to enhance knowledge transfer, stabilize clustering, improve rep quality over previous methods.
USNID [Zha+24]	Unified (Unsup. and Semi-Sup.)	Unsupervised contrastive pre-training; Use centroids from previous iteration to initialize next (centroid init strategy).	Unified framework; Centroid initialization/ or alignment for stability.	Optional; framework functions unsupervised but benefits from labels if available.	Addresses heavy reliance on labeled data, poor knowledge transfer, cluster number estimation difficulties.

Table A.3: List of expressions implying a request, used to filter sentence candidates likely containing an intent.

Expressions
please
can you
could you
would you
would it be possible to
may I
can we
could we
when
where
what
who
would it be okay if
I was wondering if
I'd appreciate it if
it would be great if
do you mind if
would you mind
I'd like to ask if
is it possible to
I'd like to request
might you
can I ask you to
let's
shall we
how about
what if we
can't we just
why don't you
maybe you could
perhaps you can
if it's not too much trouble
if you don't mind
if possible

Table A.4: Created intent taxonomy.

Label	Description
request_conference_call	A person is requesting or suggesting to have a conference call.
request_review_and_approval	A person is requesting their recipient to review and approve a document or contract.
request_update_contact_info	A person is informing their recipient(s) about changed contact information and is requesting to have them updated.
request_removal_from_list	A person is requesting to be removed from a list, often a mailing list.
request_addition_to_list	A person is requesting to be added to a list.
request_attendance_info	A person is inquiring information on whether their recipient will participate in a meeting or event.
request_availability	A person is requesting information on when a recipient(s) is available for a call or meeting, etc.
mark_calendar	A person is requesting for an event to be added to a calendar.
request_login_credentials	A person requesting login credentials such as a username and password.
request_confidentiality	A person is requesting to treat some information confidential.
request_add_cc	A person wants to be copied (CC) in an email.
propose_meeting	A person is proposing a meeting, sometimes also including the time and/or the location.
request_meeting_info	A person is requesting information on a meeting, e.g. when it takes place or where it takes place.
request_reschedule	A person is requesting to reschedule a meeting.
request_close_bug	A person is requesting a bug to be closed, as it has been resolved.
request_run_test	A person requesting their recipient to run test cases for a code or to otherwise test an application.
request_contact_information	A person is requesting contact information, including phone or cellphone numbers, fax numbers, email addresses, etc.
request_call	A person is requesting their recipient to give them a call.
request_meeting	A person is requesting to set up a meeting
request_send_document	A person is requesting to be send a document, such as a presentation, NDA, contract, spreadsheets, etc.
inform_attachment	A person is informing their recipient about an attachment included in a sent email.
request_sign_and_fax	A person is requesting for a document to be signed first and subsequently sent back via fax.
request_send_pricing_information	A person inquiring information in regards to the price for a product.
request_bug_report	A person requesting their recipient to file a bug report.
offer_assistance	A person is offering their assistance to the recipient.
request_send_feedback	A person requesting feedback, like thoughts and suggestions.
request_attendance	A person is requesting their recipient to attend a meeting or social event.
request_fax	A person is requesting their recipient to send them a fax.
request_specific_format	A person is requesting their recipient to send something in a specific format.
request_print	A person is requesting something to be print.
purchase_order	A person is requesting to issue a purchase order (PO).
request_error_details	A person is requesting information on why an error occurred.
request_reproduction_steps	A person is inquiring the reproduction steps necessary to replicate a bug or error.
request_inclusion	A person requesting another person to be included in the discussion or to a meeting.
request_urgency_with_deadline	A person is requesting something involving a deadline.
request_start_build	A person is requesting their recipient to start a build for a server.
request_create_baseline	A person is requesting their recipient to create a baseline for a server.
request_instructions_on_how_to_proceed	A person is inquiring information on the next steps for a task.
request_disregard_of_previous_request	A person is requesting their recipient to disregard a previous request or message.
request_deletion	A person is requesting the deletion of a file, email, etc.
request_visit	A person is requesting their recipient to stop by at their office to visit them.
request_access	A person is requesting access for a facility or to a file.
request_further_information	A person is requesting further information.
request_verify_information	A person is requesting to confirm a statement or to otherwise verify some information.
request_coordination	A person is requesting their recipient(s) to work with another co-worker.
request_status_updates	A person is requesting status updates on an issue and wants to be kept in the loop.
request_reminder	A person requests to be reminded of an event occurring.
request_holding_off	A person is requesting their recipient to actively delay an action.
request_follow_up	A person is requesting their recipient(s) to follow up with another person.
express_apology	A person is expressing their apologies.
express_greetings	A person is expressing greetings.
request_link	A person requests their recipient to provide them with a specific URL.
provide_link	A person is providing their recipient(s) with a specific URL.
introduction	A person is introducing a new employee.

B

FIGURES

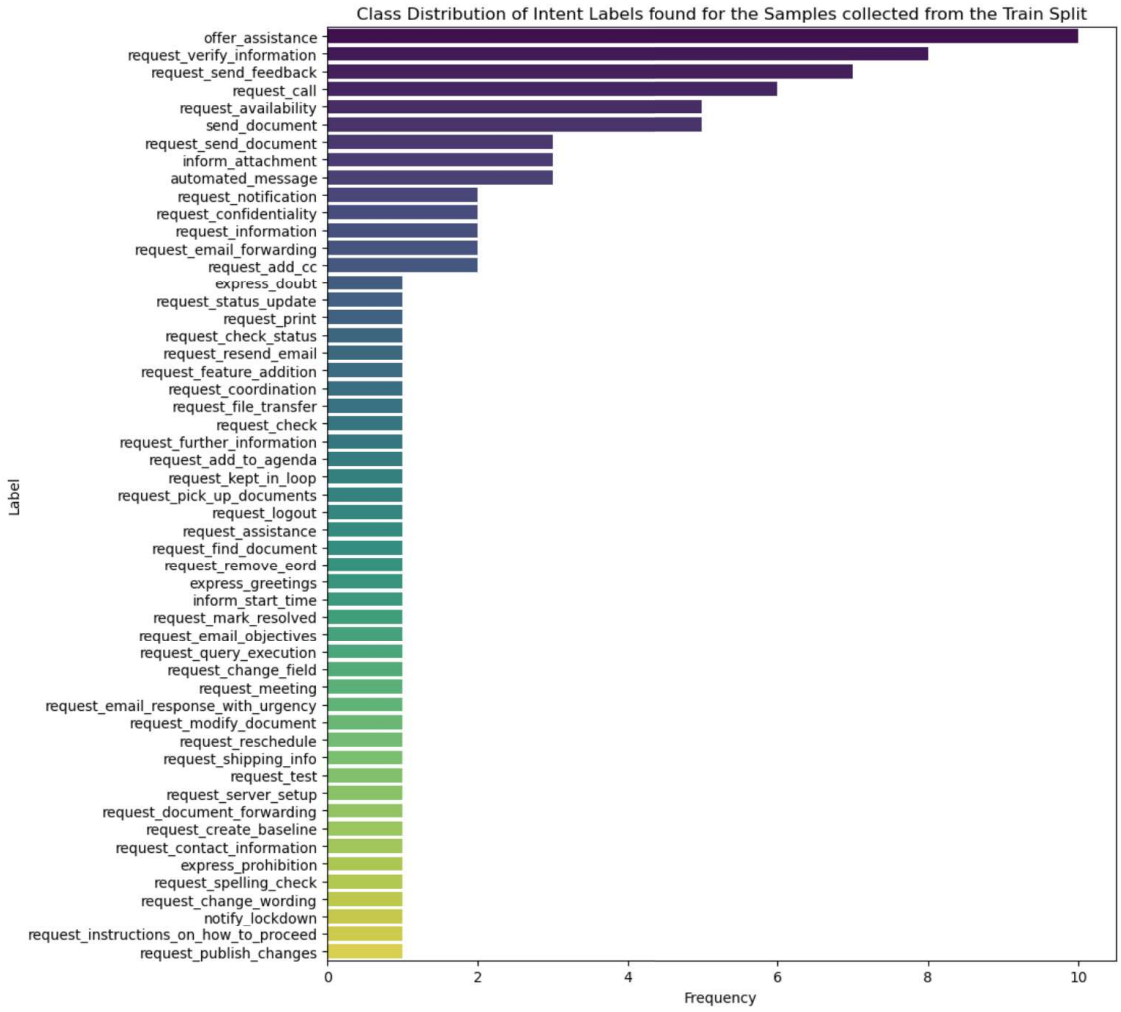

Figure B.1: Barchart of the label distributions found for the training split.

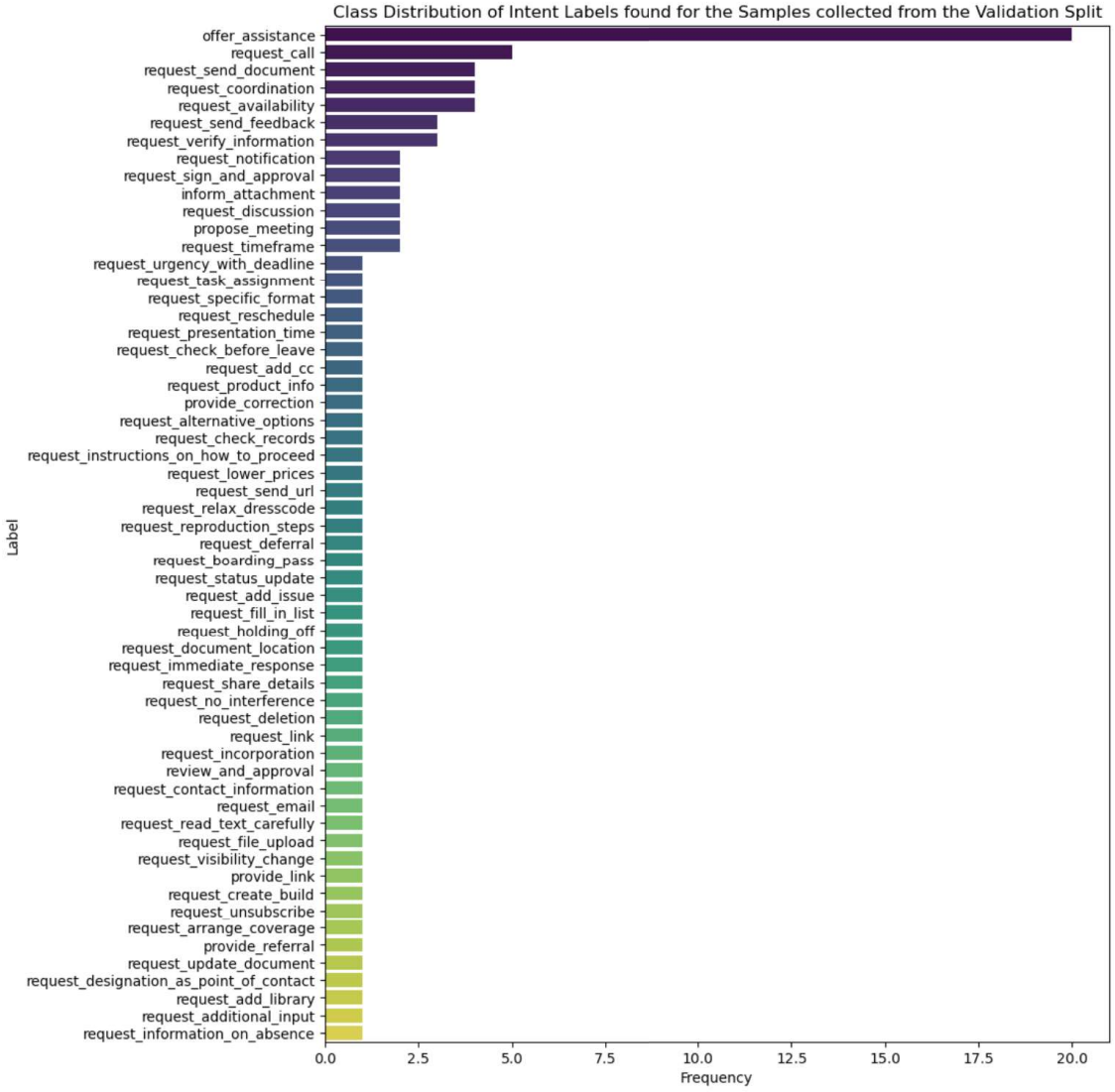

Figure B.2: Barchart of the label distributions found for the validation split.

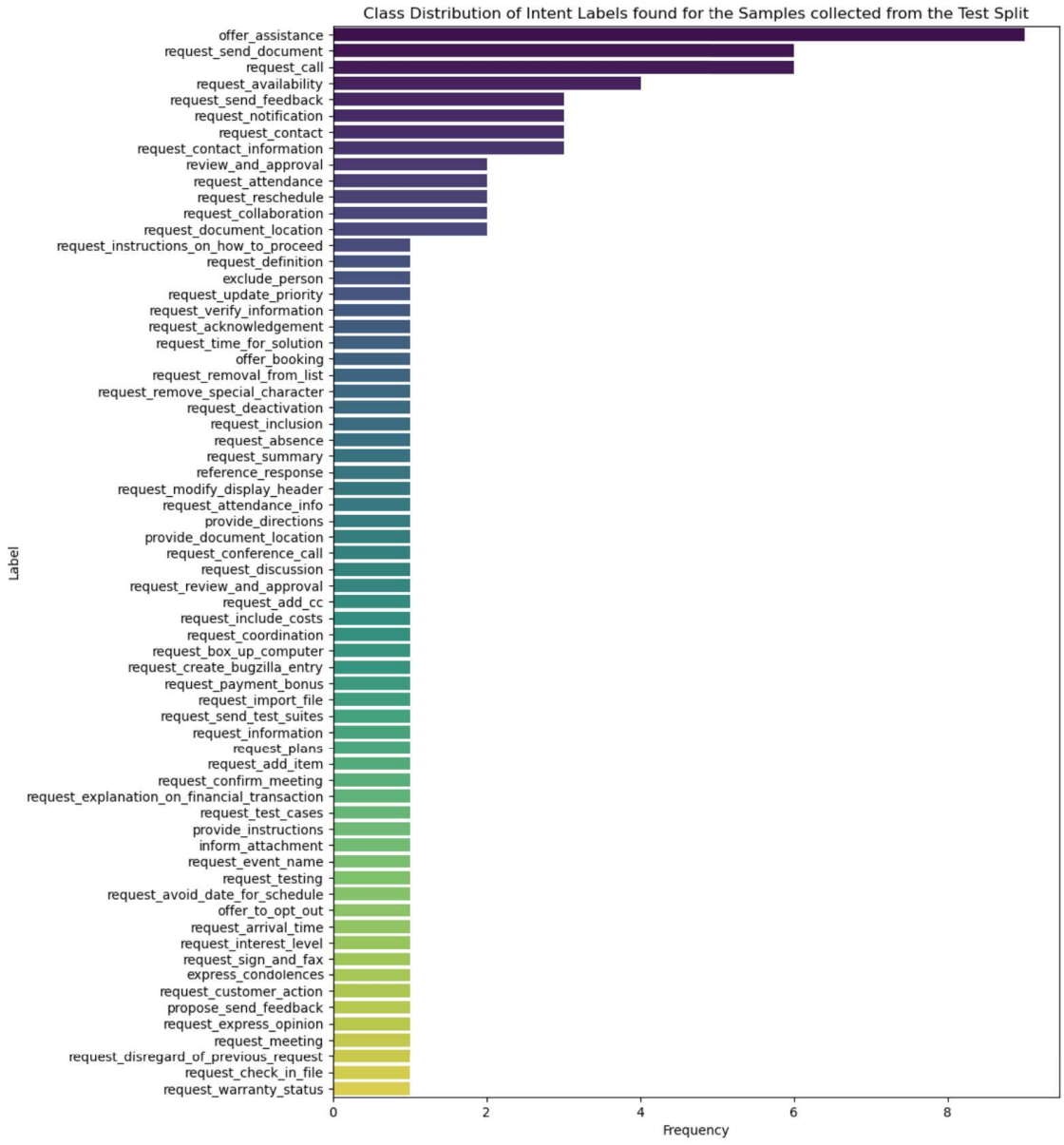

Figure B.3: Barchart of the label distributions found for the test split.

C

LISTINGS

```
You are a language model tasked with evaluating sentences for their suitability in an intent-dataset. For each given sentence, assign the following **scores** on a scale from 1 to 5, where:

- **1** = Very poor (does not meet the criterion at all)
- **5** = Excellent (fully meets the criterion)

#### Evaluation Criteria:

1. **Clarity of Intent (Intent Clarity)**
- Does the sentence clearly express an intent, such as a request, a task, or a commitment to provide information?
- Score 5 if the intent is explicit and well-articulated, with no ambiguity.

2. **Completeness of Information (Self-Containment)**
- Is the sentence complete in itself, with all necessary details provided to understand the intent without requiring additional context?
- Score 5 if the sentence provides all critical information (e.g., objects, people, tasks) needed to fully interpret the intent.

3. **Specificity (Task/Object Definition)**
- Are the task, object, or entities involved in the intent well-defined and specific?
- Score 5 if the task and any affected objects/entities are precisely described.

#### Additional Requirement:
If a sentence scores below **4** in any of these categories, briefly explain why it fails to meet the criterion.

---

#### Input Format:
Provide each sentence as a separate input. For example:
"<example obfuscated>"

#### Output Format:
For each input sentence, return the following JSON object:

{
"sentence": "<input sentence>",
"intent_clarity": <score (1-5)>,
"self_containment": <score (1-5)>,
"specificity": <score (1-5)>,
```

```
"explanation": "<brief explanation if any score lower than 4>"
}

#### Examples:

Input:
"<example obfuscated>"

Output:
{
"sentence": "<example obfuscated>",
"intent_clarity": 5,
"self_containment": 5,
"specificity": 5,
"explanation": ""
}

---

Input:
‘Sentence: <example obfuscated>‘

Output:
{
"sentence": "<example obfuscated>",
"intent_clarity": 3,
"self_containment": 2,
"specificity": 1,
"explanation": "The intent is between vague and clear, it is some form of
    commitment of deliver of information. However, ’Them’ is very vague and likely
     refers to a third party to whom the information will be delivered, but
    further context is necessary on who the information will be provided to and ’
    it’ is also very vague since it’s some form of information, however it needs
    further specification of what the information even is."
}
```

Listing C.1: Prompt used to assign quality scores for subsequent filtering

```
You are an expert in analyzing user intents from sentences, both explicit and
    implicit. When presented with a sentence, your task is to identify and
    classify the underlying intent(s) using the most appropriate and clear
    descriptors.

1. **Explicit Intent:** When the intent is directly stated, classify it using a
    precise label that clearly conveys the user’s purpose.
2. **Implicit Intent:** When the intent is not directly stated but can be inferred
     from context or common understanding, classify it based on the implied
    purpose.
3. **Avoid Ambiguity:** Choose intent descriptors that avoid vagueness or multiple
     interpretations. For example, "request_direction" is ambiguous because it
```

```
    could refer to asking for directions to a location or instructions on how to
    proceed with a task. Be mindful of this distinction.

Your goal is to ensure that the intent descriptors are unambiguous, accurate, and
    tailored to the context of the utterance.

---

#### Input Format:
"<example obfuscated>"

#### Output Format:
Return the following JSON object:

{
  "explicit_intent": "<explicit_intent>",
  "implicit_intent": "<implicit_intent>",
  "purpose": "<brief summarization of the main purpose of the utterance, focusing
      on the user's goal or desired action>"
}

#### Examples:

Input:
"<example obfuscated>"

Output:
{
  "explicit_intent": "request_call",
  "implicit_intent": "offer_help",
  "purpose": "A person is offering help in case the recipient is in need of
      assistance."
}

---

Input:
"<example obfuscated>"

Output:
{
  "explicit_intent": "request_phone_number",
  "implicit_intent": "request_phone_number",
  "purpose": "A person is requesting a phone number of an office."
}


---

Input:
```

```
"<example obfuscated>"

Output:
{
  "explicit_intent": "request_instructions",
  "implicit_intent": "request_instructions",
  "purpose": "A person is inquiring information on how to proceed."
}

---

Input:
"<example obfuscated>"

{
  "explicit_intent": "request_contact",
  "implicit_intent": "offer_assistance",
  "purpose": "A person is inviting the recipient to contact them for any further
      questions or comments, offering assistance."
}

Important: only output a valid JSON object without including anything else in your
     response like any conversational or explanatory steps, no matter what!
```

Listing C.2: Prompt used for feature generation for subsequent clustering

BIBLIOGRAPHY

[Agr+22] Monica Agrawal et al. "Large language models are few-shot clinical information extractors." In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1998–2022. DOI: 10.18653/v1/2022.emnlp-main.130. URL: https://aclanthology.org/2022.emnlp-main.130/.

[BB16] Abhijit Bendale and Terrance E Boult. "Towards open set deep networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1563–1572.

[Ben+03] Yoshua Bengio et al. "A neural probabilistic language model." In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155.

[Boj+17] Piotr Bojanowski et al. "Enriching word vectors with subword information." In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146.

[Cas+20] Iñigo Casanueva et al. "Efficient intent detection with dual sentence encoders." In: *arXiv preprint arXiv:2003.04807* (2020).

[Che+24] Ruirui Chen et al. "Is a Large Language Model a Good Annotator for Event Extraction?" In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 16. 2024, pp. 17772–17780.

[Cho+24] Juhwan Choi et al. "Multi-News+: Cost-efficient Dataset Cleansing via LLM-based Data Annotation." In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 15–29. DOI: 10.18653/v1/2024.emnlp-main.2. URL: https://aclanthology.org/2024.emnlp-main.2/.

[CCM04] William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. "Learning to Classify Email into "Speech Acts"." In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Ed. by Dekang Lin and Dekai Wu. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 309–316. URL: https://aclanthology.org/W04-3240/.

[Cou+18] Alice Coucke et al. "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces." In: *arXiv preprint arXiv:1805.10190* (2018).

[Dev+19] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

[Dou+15] Oard Douglas et al. *Avocado Research Email Collection LDC2015T03*. 2015. DOI: 10.35111/wqt6-jg60.

[E+19] Haihong E et al. "A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5467–5471. DOI: 10.18653/v1/P19-1544. URL: https://aclanthology.org/P19-1544/.

[EL20] Avia Efrat and Omer Levy. "The turking test: Can language models understand instructions?" In: *arXiv preprint arXiv:2010.11982* (2020).

[Fan+20] Lu Fan et al. "Unknown Intent Detection Using Gaussian Mixture Model with an Application to Zero-shot Intent Classification." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 1050–1060. DOI: 10.18653/v1/2020.acl-main.99. URL: https://aclanthology.org/2020.acl-main.99/.

[FL16] Geli Fei and Bing Liu. "Breaking the Closed World Assumption in Text Classification." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 506–514. DOI: 10.18653/v1/N16-1061. URL: https://aclanthology.org/N16-1061/.

[GAK23] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. "ChatGPT outperforms crowd workers for text-annotation tasks." In: *Proceedings of the National Academy of Sciences* 120.30 (July 2023). ISSN: 1091-6490. DOI: 10.1073/pnas.2305016120. URL: http://dx.doi.org/10.1073/pnas.2305016120.

[GK78] K. Chidananda Gowda and G. Krishna. "Agglomerative clustering using the concept of mutual nearest neighbourhood." In: *Pattern Recognit.* 10 (1978), pp. 105–112. URL: https://api.semanticscholar.org/CorpusID:5186751.

[Gra+24] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[Gro22] Maarten Grootendorst. *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. 2022. arXiv: 2203.05794 [cs.CL]. URL: https://arxiv.org/abs/2203.05794.

[He+24] Xingwei He et al. *AnnoLLM: Making Large Language Models to Be Better Crowdsourced Annotators*. 2024. arXiv: 2303.16854 [cs.CL]. URL: https://arxiv.org/abs/2303.16854.

[HGD90] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. "The ATIS Spoken Language Systems Pilot Corpus." In: *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*. 1990. URL: https://aclanthology.org/H90-1021/.

[Hua+23] Jiaxin Huang et al. "Large Language Models Can Self-Improve." In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 1051–1068. DOI: 10.18653/v1/2023.emnlp-main.67. URL: https://aclanthology.org/2023.emnlp-main.67/.

[Kim+22] Hyuhng Joon Kim et al. *Self-Generated In-Context Learning: Leveraging Auto-regressive Language Models as a Demonstration Generator*. 2022. arXiv: 2206.08082 [cs.CL]. URL: https://arxiv.org/abs/2206.08082.

[KY04] Bryan Klimt and Yiming Yang. "The enron corpus: a new dataset for email classification research." In: *Proceedings of the 15th European Conference on Machine Learning*. ECML'04. Pisa, Italy: Springer-Verlag, 2004, pp. 217–226. ISBN: 3540231056. DOI: 10.1007/978-3-540-30115-8_22. URL: https://doi.org/10.1007/978-3-540-30115-8_22.

[Lar+19] Stefan Larson et al. "An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1311–1316. DOI: 10.18653/v1/D19-1131. URL: https://aclanthology.org/D19-1131/.

[Li+23] Minzhi Li et al. "CoAnnotating: Uncertainty-Guided Work Allocation between Human and Large Language Models for Data Annotation." In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023. DOI: 10.18653/v1/2023.emnlp-main.92. URL: http://dx.doi.org/10.18653/v1/2023.emnlp-main.92.

[Li+21] Yunfan Li et al. "Contrastive clustering." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 10. 2021, pp. 8547–8555.

[LX19] Ting-En Lin and Hua Xu. "Deep Unknown Intent Detection with Margin Loss." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5491–5496. DOI: 10.18653/v1/P19-1548. URL: https://aclanthology.org/P19-1548/.

[LXZ19] Ting-En Lin, Hua Xu, and Hanlei Zhang. *Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement*. 2019. arXiv: 1911.08891 [cs.CL]. URL: https://arxiv.org/abs/1911.08891.

[Mac+67] James MacQueen et al. "Some methods for classification and analysis of multivariate observations." In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[McI+18] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection." In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861. URL: https://doi.org/10.21105/joss.00861.

[Mik+13] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL]. URL: https://arxiv.org/abs/1301.3781.

[Ope+24] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162/.

[Pet+18] Matthew E. Peters et al. "Deep Contextualized Word Representations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202/.

[Qin+19] Libo Qin et al. "A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2078–2087. DOI: 10.18653/v1/D19-1214. URL: https://aclanthology.org/D19-1214/.

[Raf+20] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." In: *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435.

[RG19] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by

Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410/.

[Sap+16] Maya Sappelli et al. "Assessing e-mail intent and tasks in e-mail messages." In: *Information Sciences* 358 (2016), pp. 1–17.

[Sch+13] Walter J. Scheirer et al. "Toward Open Set Recognition." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.7 (2013), pp. 1757–1772. DOI: 10.1109/TPAMI.2012.256.

[SXL17] Lei Shu, Hu Xu, and Bing Liu. "DOC: Deep Open Classification of Text Documents." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2911–2916. DOI: 10.18653/v1/D17-1314. URL: https://aclanthology.org/D17-1314/.

[Tan+24] Zhen Tan et al. "Large language models for data annotation and synthesis: A survey." In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 930–957.

[Tea+24] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[Tse+24] Yu-Min Tseng et al. *Are Expert-Level Language Models Expert-Level Annotators?* 2024. arXiv: 2410.03254 [cs.CL]. URL: https://arxiv.org/abs/2410.03254.

[Tun+22] Lewis Tunstall et al. *Efficient Few-Shot Learning Without Prompts*. 2022. arXiv: 2209.11055 [cs.CL]. URL: https://arxiv.org/abs/2209.11055.

[Vas17] A Vaswani. "Attention is all you need." In: *Advances in Neural Information Processing Systems* (2017).

[Wan+24a] Mengting Wan et al. "TnT-LLM: Text Mining at Scale with Large Language Models." In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '24. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 5836–5847. ISBN: 9798400704901. DOI: 10.1145/3637528.3671647. URL: https://doi.org/10.1145/3637528.3671647.

[Wan+19] Wei Wang et al. "Context-Aware Intent Identification in Email Conversations." In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'19. Paris, France: Association for Computing Machinery, 2019, pp. 585–594. ISBN: 9781450361729. DOI: 10.1145/3331184.3331260. URL: https://doi.org/10.1145/3331184.3331260.

[Wan+24b] Xinru Wang et al. "Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels." In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: 10.1145/3613904.3641960. URL: https://doi.org/10.1145/3613904.3641960.

[XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis." In: *International conference on machine learning*. PMLR. 2016, pp. 478–487.

[Xu+15] Jiaming Xu et al. "Short Text Clustering via Convolutional Neural Networks." In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Ed. by Phil Blunsom et al. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 62–69. DOI: 10.3115/v1/W15-1509. URL: https://aclanthology.org/W15-1509/.

[YCS24] Sachin Yadav, Tejaswi Choppa, and Dominik Schlechtweg. *Towards Automating Text Annotation: A Case Study on Semantic Proximity Annotation using GPT-4*. 2024. arXiv: 2407.04130 [cs.CL]. URL: https://arxiv.org/abs/2407.04130.

[Yan+17] Bo Yang et al. "Towards K-means-friendly spaces: simultaneous deep learning and clustering." In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 3861–3870.

[Zha+18] Chenwei Zhang et al. "Joint Slot Filling and Intent Detection via Capsule Neural Networks." In: *CoRR* abs/1812.09471 (2018). arXiv: 1812.09471. URL: http://arxiv.org/abs/1812.09471.

[Zha+21a] Dejiao Zhang et al. "Supporting Clustering with Contrastive Learning." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 5419–5430. DOI: 10.18653/v1/2021.naacl-main.427. URL: https://aclanthology.org/2021.naacl-main.427/.

[ZXL21] Hanlei Zhang, Hua Xu, and Ting-En Lin. "Deep Open Intent Classification with Adaptive Decision Boundary." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.16 (May 2021), pp. 14374–14382. ISSN: 2159-5399. DOI: 10.1609/aaai.v35i16.17690. URL: http://dx.doi.org/10.1609/aaai.v35i16.17690.

[Zha+21b] Hanlei Zhang et al. "Discovering New Intents with Deep Aligned Clustering." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.16 (May 2021), pp. 14365–14373. ISSN: 2159-5399. DOI: 10.1609/aaai.v35i16.17689. URL: http://dx.doi.org/10.1609/aaai.v35i16.17689.

[Zha+21c] Hanlei Zhang et al. "TEXTOIR: An Integrated and Visualized Platform for Text Open Intent Recognition." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Ed. by Heng Ji, Jong C. Park, and Rui Xia. Online: Association for Computational Linguistics, Aug. 2021, pp. 167–174. DOI: 10.18653/v1/2021.acl-demo.20. URL: https://aclanthology.org/2021.acl-demo.20/.

[Zha+24] Hanlei Zhang et al. "A Clustering Framework for Unsupervised and Semi-Supervised New Intent Discovery." In: *IEEE Transactions on Knowledge and Data Engineering* 36.11 (Nov. 2024), pp. 5468–5481. ISSN: 2326-3865. DOI: 10.1109/tkde.2023.3340732. URL: http://dx.doi.org/10.1109/TKDE.2023.3340732.

[Zha+22a] Yuwei Zhang et al. *New Intent Discovery with Pre-training and Contrastive Learning*. 2022. arXiv: 2205.12914 [cs.CL]. URL: https://arxiv.org/abs/2205.12914.

[Zha+22b] Mengjie Zhao et al. "LMTurk: Few-Shot Learners as Crowdsourcing Workers in a Language-Model-as-a-Service Framework." In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 675–692. DOI: 10.18653/v1/2022.findings-naacl.51. URL: https://aclanthology.org/2022.findings-naacl.51/.

[Zho+22] Yongchao Zhou et al. "Large language models are human-level prompt engineers." In: *The Eleventh International Conference on Learning Representations*. 2022.