UNIVERSITÀ DI PISA

DIPARTIMENTO DI INFORMATICA

Laurea Magistrale in Data Science and Business Informatics

Classification for Joint Search and Recommendation

Relatore:

Prof: Raffaele Perego

Relatore:

Prof: Franco Nardini

Correlatore:

Cristina-Ioana Muntean

Candidato:

Ajla Rustemi

ANNO ACCADEMICO 2024/2025

First, I wish to thank my supervisor, Raffaele Perego, for the opportunity to work on this project and for his expert mentorship. I am equally grateful to my co-supervisor, Cristina Ioana Muntean, and to Professor Franco Nardini; their guidance and patience were necessary in bringing this work to completion.

On a personal note, I am forever indebted to my family for their constant presence and for helping me overcome every obstacle along the way. I also want to dedicate a special thanks to my close friends. You have created a second home for me here, providing the strength and joy I needed to finish this chapter of my life. Thank You!

Abstract

As conversational systems evolve, we expect conversational systems to act as partners that can smoothly switch between answering our questions and suggesting the right products. This integration, known as Joint Conversational Search and Recommendation (CSR), faces a critical architectural hurdle; intent routing problem. This thesis explores the challenge of determining whether a user needs open-domain information or a personalized suggestion, using the CoSRec dataset as our primary benchmark. Our initial experiments reveal a persistent struggle for models to distinguish between exploratory and transactional intents. In trying to solve this, we uncovered a counter-intuitive Profile Paradox. While a user's personal history is essential for ranking items, injecting it too early into the intent router actually blinds the system. In Search-priority setups, these personalization signals swamped the immediate conversational context, causing the system to ignore what the user was actually saying and default to recommendation behavior. This led to a performance near failure, with discriminative ability (ROC-AUC) dropping to a near-zero 6.1%. We identify the root of this failure as a Semantic Asymmetry where recommendation language acts as a dominant trait, masking recessive search signals. By applying LLM-driven data augmentation to strengthen the search signal and ensure that it was not drowned out by recommendation data, we successfully executed a Signal Recovery strategy, restoring F1-scores to a robust 89.9%. Finally, this work argues for a separation to stay precise and maintain accuracy, intent routing must remain context-only, reserving personalization strictly for the downstream ranking stage where it belongs.

Contents

1 Introduction **5**
- 1.1 The Evolution of Information Access . . . 5
  - 1.1.1 The Conversational Revolution . . . 6
  - 1.1.2 Joint Conversational Search and Recommendation . . . 6
  - 1.1.3 The Motivation and Proposed Solution: The Chicken-and-Egg Dilemma 7

2 Related Work **9**
- 2.1 Foundations of Search and Recommendation . . . 9
  - 2.1.1 Information Retrieval . . . 9
  - 2.1.2 Recommendation Systems . . . 13
  - 2.1.3 Conversational Search . . . 17
  - 2.1.4 Conversational Recommendation . . . 18
- 2.2 Joint Conversational Search and Recommendation . . . 19
  - 2.2.1 System Perspective . . . 19
  - 2.2.2 Data and Evaluation Challenges . . . 20
- 2.3 Semantic ambiguity on intent classification . . . 21

3 Problem Description and Setup **23**
- 3.1 Dataset Overview . . . 23
  - 3.1.1 Dataset Composition . . . 24
  - 3.1.2 Structure of a Conversation . . . 26
  - 3.1.3 Intent Taxonomy . . . 28
  - 3.1.4 Data Sources . . . 31
  - 3.1.5 Annotation Scheme . . . 33
  - 3.1.6 Dataset Statistics and Analysis . . . 33
- 3.2 Problem Definition and Setup . . . 36
  - 3.2.1 The Intent Routing Task . . . 36
  - 3.2.2 Models for Intent Routing . . . 37
  - 3.2.3 Training Objective and Loss Functions . . . 38

4 Methodology **40**
- 4.1 Label Distribution . . . 40
  - 4.1.1 Utterance Sparsity and Structure . . . 42
- 4.2 Multi-label Formulation . . . 46
  - 4.2.1 Results Across Training Data Configurations . . . 47

4.2.2 Feature Analysis for Classical Models . . . 48
4.3 Transformer-Based Intent Classification . . . 51
4.3.1 Training phase . . . 52
4.4 Binary Intent Routing Methodology . . . 55
4.4.1 Model Architecture . . . 57
4.4.2 Optimization and Training Strategy . . . 58

5 Experiments Results and Analysis 60
5.1 Overall Results . . . 60
5.1.1 Per-label Performance and Error Analysis . . . 61
5.1.2 Probabilistic Metrics (AUC/AP) . . . 62
5.2 The Personalization Paradox in Intent Routing . . . 63
5.2.1 Personalization Bias and Systematic Failure . . . 64
5.2.2 Parameter Sensitivity and Learning Dynamics . . . 64
5.3 Ambiguity Analysis: Lexical and Probabilistic Asymmetry . . . 65
5.3.1 Probabilistic Behaviour on Mixed-Intent Utterances . . . 66
5.3.2 Lexical Drivers of Asymmetry . . . 67
5.4 Strategic Routing Policies . . . 70
5.4.1 Recommendation-Priority Policy (Model A) . . . 70
5.4.2 Search-Priority Policy (Model B) . . . 77
5.5 Reformulating Intent Routing: From Multi-label to Binary . . . 84
5.5.1 Dataset Distribution . . . 84
5.5.2 Analysis of Intent Classification . . . 85
5.5.3 Mixed Intents in CSR . . . 86
5.6 Data Augmentation and Signal Recovery . . . 88
5.6.1 Augmentation . . . 89

6 Future work and Conclusions 91
6.1 Conclusions . . . 91
6.2 Future Work . . . 92

Chapter 1

Introduction

1.1 The Evolution of Information Access

In the field of Information Retrieval (IR), the foundational framework has always rested on three pillars: the document, the query, and the search results. While these components remain constant, the methods we use to navigate them have shifted dramatically alongside the broader evolution of computing.

In the early stages of the digital era, computing was primarily a tool for data processing rather than "knowledge" discovery. Information was largely synonymous with structured records, neatly organized rows and columns housed within relational databases. The idea was to keep things fast, consistent, and correct. Despite their effectiveness, these systems were not very helpful in terms of context or interpretation. The problem? They only handled raw data. There was not much meaning attached, nothing tailored for real human understanding or usefulness. [2]. This limitation came as a result of the unit of processing being raw data, which lacked the human centric enrichment of meaning and utility that later information systems sought to provide.

Early chatbots and conversational tools ran into similar roadblocks, that worked with strict scripts and decision trees. Developers had to try to predict everything a user might say and that got messy fast, especially because real conversations are unpredictable and full of ambiguity [5]. These types of systems relied heavily on pattern matching, skipping over deeper issues such as syntax, semantics, or pragmatics. They could not track conversation threads or handle references that traversed multiple messages. So, most of the time, they just kept structured records tidy instead of adapting to what users actually wanted or needed in that moment.

As information systems moved forward, the focus shifted from plain data to information data that's been given meaning and made useful for people [2]. This shift powered the rise of classical IR systems. Now, the goal was not just to store data, but to pull relevant documents from massive, unstructured collections when someone asked for them. Ranking algorithms, relevance feedback, and big evaluation projects like Text REtrieval Conference (TREC) became the norm and helped turn IR into a major research field [30, 3].

More recently, the spotlight has swung towards knowledge: information that's organized, structured, and aware of context and intent. Knowledge-driven systems do not just find

documents, they help people make decisions, explain things, and synthesize new ideas [4]. While knowledge has to live in people's memory or on paper, now we're organizing it at scale with the web, knowledge graphs, and neural models. That's completely changed how we access information. The trajectory has gone from crunching raw data to building systems that care about intent, context, and synthesis. Conversational systems are the most expressive example of this change [12].

1.1.1 The Conversational Revolution

Lately, conversational agents have taken over as the go-to way to access information. Instead of wrestling with clunky query languages, people just chat with these systems in plain language. That makes things a lot more approachable especially when you don't know exactly what you're looking for or when your needs are complex. [33].

Unlike old-school search engines that just returns results after a single query, conversational agents keep the conversation going over multiple turns. They have to remember what's been said, keep track of the conversation's state, and figure out what you mean even when you are not being super clear. This is especially important when users refine their requests step by step. [34].

Within conversational information access, two big camps have taken shape. On one side, there's Conversational Search, helping users explore open-ended questions through dialogue. On the other, there is Conversational Recommendation guiding users to specific items by learning about their preferences [53]. Although both paradigms rely on shared components such as natural language understanding and context tracking, they have evolved largely in isolation.

As a result, existing systems are often highly optimized for a single interaction mode, but poorly equipped to handle conversational scenarios where informational and recommendation-oriented needs coexist or shift dynamically.

1.1.2 Joint Conversational Search and Recommendation

In real world interactions, people do not stick to just searching or just asking for recommendations. The line between the two is blurry. Users may begin with an informational query and later request suggestions, or start off with a recommendation-oriented request that requires more information or factual clarification. Researches have pointed out that conversational interaction enables systems to actively clarify user needs, rather than assuming a fixed intent for the whole session. [53].

This observation motivates the paradigm of Joint Conversational Search and Recommendation (CSR), which seeks to unify **conversational search** and **conversational recommendation** within a single framework. CSR systems must dynamically alternate between retrieving open-domain information and providing personalized item suggestions, depending on the evolving conversational context.

Despite growing interest, CSR remains underexplored due to both methodological and data related challenges. Most existing conversational datasets and benchmarks focus on either search or recommendation in isolation, limiting so the study of mixed-intent interactions [33]. Furthermore, prior work on personalization has shown that not all queries benefit

equally from user-specific signals, and that inappropriate personalization can degrade performance [47].

Figuring out intent routing is at the heart of the CSR challenge. Every time a user says something, the system has to make a call: should it send the request to a search engine or to a recommendation module? If it picks wrong here, the whole experience can spiral and that's why it's so important to have good datasets and a way to model intent ambiguity directly. This is what drives the focus of this thesis.

1.1.3 The Motivation and Proposed Solution: The Chicken-and-Egg Dilemma

The real challenge is a classic chicken-and-egg scenario. Progress on joint Conversational Search and Recommendation (CSR) keeps stalling because there just are not enough public datasets that let researchers evaluate both search and recommendation together. Traditional Information Retrieval (IR) datasets are typically constructed through relevance judgments over open-domain corpora, while recommended systems rely on large-scale interaction logs collected from deployed platforms [10]. These two approaches do not mix, and that is what prevents shared benchmarks for true conversational systems.

Although recent work has highlighted the growing importance of conversational information access [33, 34], most existing resources focus exclusively on either conversational search or conversational recommendation. As a result, systems are often tested under the assumptions that despite evidence that real user interactions frequently involve mixed or evolving intents [53]. The gap between how we test systems and how people actually use them makes it harder to build conversational models that can adapt on the fly.

This thesis tackles that gap with CoSRec, the first dataset made for joint evaluation of conversational search and recommendation. CoSRec brings together about 9,000 conversations, including pure search, pure recommendation, and chats where users switch between the two. It even comes with separate ground truths for each mode. By separating intent routing from the actual ranking downstream, CoSRec lets us experiment and see how well a system can figure out what the user really wants especially when it's not clear.

The main bottleneck here is intent routing. That's the point where the system decides: does this user utterance call for a search, or a recommendation? One particular issue is the "Profile Paradox." If a system leans on user profiles too early, it tends to over-recommend and ends up ignoring when the user just wants to search. Earlier research shows that too much personalization can actually make things worse if the system focuses too much on the user and not enough on the current query, performance drops [47].

These challenges reflect back at a chicken-and-egg dilemma in CSR research: integrated conversational systems are required to generate realistic interaction logs, yet such systems cannot be reliably developed or evaluated without the access to those logs. Proprietary data makes this even trickier. This thesis addresses the fundamental 'chicken-and-egg' dilemma of joint Conversational Search and Recommendation (CSR) by isolating intent routing as a standalone, measurable component.

To resolve this circular dependency, this thesis provides a three-fold contribution. First, we introduce the CoSRec dataset, the first benchmark for joint CSR synthesized through

a grounded LLM-driven framework (LLM-CSSS) to overcome initial data sparsity. Second, our research uncovers a fundamental semantic asymmetry providing a probabilistic deep dive proving that search intents are linguistically "fragile" compared to "loud" recommendation cues, while identifying the "Profile Paradox" where early-stage personalization actually biases the system toward failure. Finally, we demonstrate a successful "Signal Recovery" strategy using targeted data augmentation, proving that a DeBERTa-based router can transform an initially failing search-priority policy into a robust operational gatekeeper for integrated conversational agents.

Chapter 2

Related Work

2.1 Foundations of Search and Recommendation

Information Retrieval (IR) and Recommendation Systems represent two fundamental paradigms for accessing information and items from large collections. While both aim to provide relevant results to users, they differ in how user intent is expressed and modeled. IR systems rely on explicit user queries to retrieve relevant documents, whereas recommendation systems infer user preferences from implicit signals such as interaction history. Traditionally, these paradigms have been studied separately, with distinct modeling approaches and evaluation frameworks.

However, modern user interactions, especially in conversational settings, often combine information-seeking and preference-driven behaviors and this motivates the need to understand both paradigms in parallel before analyzing their integration in joint conversational systems.

In the following sections, we first review the foundations of Information Retrieval, then discuss Recommendation Systems, and finally examine their convergence in conversational settings.

2.1.1 Information Retrieval

Information Retrieval (IR) focuses on retrieving relevant documents in response to an explicit user query. Early IR systems assumed that users could clearly articulate their information needs using keywords or structured queries. The system's primary responsibility was to retrieve and rank documents according to their estimated relevance to the query.

Over time, IR research has focused on ranking models, relevance estimation, and evaluation metrics such as precision and recall. Classical approaches include Boolean retrieval, vector-space models, and probabilistic methods, while more recent developments incorporate neural and representation-based models.

Despite these advances, IR systems fundamentally rely on explicit queries, assuming that user intent is clearly specified. This assumption becomes a limitation in conversational settings, where user intent may be ambiguous or evolve over time.

Recommendation Systems

Recommendation systems aim to suggest relevant items by modeling user preferences, interaction histories, and contextual signals. Unlike IR systems, which rely on explicit queries, recommendation systems operate on implicit feedback, such as browsing behavior or past interactions.

These systems emerged from the observation that users often struggle to specify their needs explicitly, particularly in domains such as e-commerce or media consumption. Instead of responding to queries, recommender systems proactively infer latent preferences and generate personalized suggestions.

Common approaches include collaborative filtering, content-based filtering, and hybrid models. More recent work incorporates neural architectures to capture complex user–item relationships. As a result, recommendation systems are inherently personalized, with relevance defined relative to individual users.

From Separation to Convergence

For many years, information retrieval and recommendation systems evolved largely in isolation. IR research focused on query–document relevance and ranking, while recommendation research emphasized personalization and user preference modeling. This separation was reflected in datasets, evaluation protocols, and modeling assumptions.

However, modern user interactions increasingly blur the boundary between these paradigms. In conversational settings, users rarely operate in a purely search-only or recommendation-only mode. Instead, interactions often involve a mixture of informational queries, preference expressions, and decision-making steps.

This convergence has led to growing interest in joint conversational search and recommendation systems, which aim to integrate information-seeking and recommendation behaviors within a unified framework.

Early Developments

Information Retrieval (IR) emerged in response to the need for organizing and accessing large document collections. Early systems relied on keyword-based matching, most notably the Boolean retrieval model, where documents were retrieved if they satisfied logical query expressions. While simple, this approach lacked ranking capabilities and often produced large, unstructured result sets.

To address these limitations, the vector space model (VSM) [44] introduced a similarity-based framework in which documents and queries are represented as vectors in a high-dimensional space. Relevance is estimated using measures such as cosine similarity, enabling ranked retrieval.

This approach also enabled term weighting schemes such as TF–IDF [43], which capture the importance of terms within documents relative to the entire collection. The VSM marked a transition from exact matching to approximate matching and remains foundational in modern retrieval systems.

Evolution of Probabilistic Retrieval Models

Probabilistic models further advanced IR by framing retrieval as the estimation of the probability that a document is relevant to a query. This idea was formalized in the Probability Ranking Principle (PRP) [39], which states that documents should be ranked by decreasing probability of relevance:

$$\text{rank documents by } P(R = 1 \mid d, q)$$

Building on this principle, models such as BM25 [40] introduced term frequency saturation and document length normalization, becoming a widely used baseline.

Relevance feedback methods, including Rocchio's algorithm [42], further improved retrieval by refining queries based on user feedback, marking an early step toward interactive and user-centered retrieval.

Toward User Modeling and Context-Aware Retrieval

These developments shifted IR toward a more user-centered paradigm, incorporating interaction signals such as query reformulation and session behavior. However, classical IR systems typically assume a single, well-defined information need per query.

This assumption becomes limiting in conversational settings, where user intent is often ambiguous and evolves over time. These limitations motivated the transition toward learning-based and neural retrieval approaches, which aim to better capture semantic relationships and dynamic intent.

Advancements in Information Retrieval

As IR evolved, two key directions emerged: the integration of Natural Language Processing (NLP) techniques and the adoption of machine learning for relevance estimation.

Natural Language Processing in Information Retrieval

Early IR models ignored linguistic structure, leading to issues such as vocabulary mismatch and ambiguity. NLP techniques, including stemming and semantic representations, enabled systems to move beyond keyword matching. Manning et al. [29] highlighted the importance of modeling meaning, leading to semantic search approaches that retrieve documents based on inferred relevance.

These advances also improved user intent understanding, particularly in conversational contexts.

Machine Learning in Information Retrieval

Machine learning introduced data-driven relevance modeling, replacing manually designed ranking functions. A key development is pairwise learning-to-rank, exemplified by RankNet [**?**], which optimizes document ordering:

$$\mathcal{L}_{\text{RankNet}} = -\log \sigma \left(f(q, d^+) - f(q, d^-) \right)$$

These methods enabled the integration of diverse signals, including user behavior, although early approaches remained limited by feature engineering.

From Linguistic Modeling to Data-Driven Semantics

The combination of NLP and machine learning transformed IR into a semantic, data-driven discipline. These developments are particularly relevant for conversational systems, where user intent is implicit, evolving, and closely tied to context and personalization.

Dense Retrieval Techniques

Introduction to Dense Retrieval

Dense retrieval represents a shift from sparse lexical matching to semantic representation learning. Traditional IR systems rely on exact term overlap, which limits their ability to handle paraphrasing and contextual variation, particularly in short or conversational queries.

In contrast, dense retrieval maps queries and documents into a shared continuous vector space, where semantic similarity is captured through geometric proximity. Neural models, typically transformer-based, encode text such that semantically related inputs are close even without lexical overlap.

While early ideas emerged from latent semantic indexing [16] and word embeddings [32], dense retrieval became practical with large pretrained language models.

Sparse vs. Dense Retrieval

Sparse methods such as TF–IDF and BM25 [41] rely on lexical overlap and are highly interpretable and efficient, but struggle with semantic mismatch. Dense retrieval, by contrast, captures semantic and contextual information, enabling queries such as "comfortable shoes for long walks" to retrieve documents describing "ergonomic footwear" even without shared terms.

$$\mathrm{sim}(q, d) = \mathbf{h}_q^\top \mathbf{h}_d$$

However, dense retrieval introduces trade-offs, including higher computational cost and reduced interpretability. As a result, many systems adopt hybrid approaches combining sparse and dense signals [28].

Neural Retrieval Models

The effectiveness of dense retrieval is closely tied to transformer-based architectures such as BERT [17], which provide rich contextual representations. Most systems adopt a dual-encoder architecture, where queries and documents are encoded independently and compared via similarity functions.

A prominent example is Dense Passage Retrieval (DPR) [23], which uses contrastive learning to align query and document representations. DPR demonstrated that dense retrieval can outperform BM25 on semantic retrieval tasks, particularly for complex queries.

Relevance to This Work

While dense retrieval models such as DPR have significantly improved semantic matching, they focus on *how* to retrieve relevant information rather than *when* retrieval should be performed.

In joint Conversational Search and Recommendation (CSR) systems, this distinction is critical, retrieval must be guided by intent routing, particularly in scenarios where user intent is ambiguous or evolves over time. This thesis addresses this gap through the CoSRec framework, focusing on how retrieval interacts with intent ambiguity and system-level routing policies.

2.1.2 Recommendation Systems

Recommendation systems have become a core component of modern information access platforms, playing a crucial role in guiding users through large and complex information spaces. They are widely deployed in domains such as e-commerce, multimedia streaming, news delivery, and social media, these systems solve a major problem: the amount of data is so huge that users can no longer browse through it all manually. By proactively suggesting items that align with a user's preferences, recommendation systems aim to improve user satisfaction, increase engagement, and support decision-making processes.

Early work in this area emphasized the importance of personalization as a distinguishing feature of recommendation systems. Resnick and Varian (1997) were among the first to formalize recommender systems as a distinct research area, highlighting their potential to filter information based on user preferences rather than relying solely on explicit queries, [37]. This paradigm shift marked a departure from traditional information retrieval (IR), where users are assumed to know what they are searching for and can express it through keywords or structured queries.

Collaborative Filtering

Collaborative filtering (CF) is one of the earliest and most influential approaches to recommendation. The central assumption underlying CF is that users who have reveal similar behavior in the past, such as rating, purchasing, or consuming similar items, are likely to have similar preferences in the future. CF methods operate by leveraging patterns in user–item interaction data, without requiring explicit knowledge of item content.

Early systems relied on 'memory-based' math to find patterns. They compared users to see who had similar tastes, or compared products to see which ones were often bought together. These methods were very effective at first, but they became difficult to manage as the number of users and items grew too large for the computer to handle quickly. [45].

A major advance in collaborative filtering came with the introduction of model-based approaches, particularly matrix factorization techniques. This paper, [24], showed that inactive factor models, which represent users and items in a shared low-dimensional space, could capture complex preference patterns while scaling efficiently to large datasets.In this space, user-item interactions are modeled as inner products of vectors.

$$\hat{r}_{ui} = q_i^T p_u$$

Where:

$\hat{r}_{ui}$ is the predicted rating of user u for item i.

$q_i \in \mathbb{R}^f$ is the latent factor vector for the item.

$p_u \in \mathbb{R}^f$ is the latent factor vector for the user. Matrix factorization became the dominant paradigm in recommender systems research and was famously adopted in the Netflix Prize competition.

Despite their success, collaborative filtering methods suffer from well-known limitations. The cold-start problem arises when new users or items lack sufficient interaction history, making reliable recommendations difficult. Additionally, CF methods are vulnerable to data sparsity, as real-world user–item interaction matrices are often extremely sparse and these limitations have motivated the exploration of alternative approaches that rely less heavily on historical interaction data.

Content-Based Recommendation

Content-based systems solve some of the problems of traditional methods by looking at the items themselves rather than other people. Instead of worrying about what your neighbors like, these systems suggest things that match your specific taste. For example, if you like action movies starring a certain actor, the system looks at movie descriptions and 'tags' to find more films just like those.

Early content-based systems employed techniques from information retrieval, such as TF–IDF representations and cosine similarity, to match user profiles with item descriptions ([44, 29]). In this sense, content-based recommendation can be viewed as a personalized form of IR, where the query is implicitly derived from the user's past behavior.

While content-based approaches are more robust to cold-start issues for items and offer greater interpretability, they also introduce new challenges. Over-specialization is a common problem: users may be repeatedly exposed to items that are too similar to their past choices, limiting discovery and diversity. Furthermore, constructing rich and accurate item representations can be difficult, particularly in domains where content features are noisy, unstructured, or subjective.

Hybrid Recommendation Systems

Hybrid recommendation systems bridge the gap between collaborative and content-based methods. Instead of relying on just one source of info, they pull together user habits, product details, and even external context to make better guesses

Hybridization strategies vary widely, ranging from simple linear combinations of CF and content-based scores to more sophisticated as high-tech AI that learns from different data types simultaneously [11]. Studies have shown that hybrid systems often outperform single-method approaches, especially when the system does not have much data to work with yet.

The rise of deep learning has further accelerated the development of hybrid recommender systems. Neural architectures can naturally integrate heterogeneous data sources, such as text, images, and user metadata, enabling richer user and item representations [50]. These models erase the line between 'looking at people' and 'looking at items.' Instead, they treat

the whole problem as a single task: teaching the AI how to understand the relationship between everything in its database.

Recommendation Systems and User Modeling

A defining characteristic of recommendation systems, distinguishing them from classical IR systems, is the central role of user modeling, while IR traditionally treats queries as independent events, recommender systems maintain persistent representations of users that evolve over time. These representations may encode long-term preferences, short-term interests, demographic attributes, or contextual factors.

User modeling introduces both opportunities and challenges. On the one hand, personalization enables highly tailored recommendations that adapt to individual users. On the other hand, it raises concerns related to bias, fairness, and interpretability. User profiles can boost existing preferences, reinforce filter bubbles, and introduce systematic biases into recommendation outcomes [35].

These issues become particularly important in conversational and interactive settings, where user preferences are refined in real-time through dialogue. In such scenarios, recommendation systems must not only infer preferences from historical data but also interpret natural language utterances and conversational context a challenge that closely aligns recommendation research with developments in conversational information retrieval.

From Recommendation to Conversational Recommendation

More recent work has extended traditional recommendation systems into the conversational domain. Conversational Recommendation Systems (CRS) engage users in multi-turn dialogues, allowing preferences to be elicited, refined, and updated interactively [54]. Unlike static recommenders, CRS must decide when to ask clarifying questions, when to present recommendations, and how to balance exploration and exploitation.

Existing CRS datasets and models typically assume a closed-world setting, where the system recommends items from a fixed catalogue and relies heavily on personalization signals such as user profiles or past interactions [26, 55]. Because of this, researchers in search and recommendations have mostly worked in their own separate worlds and this is surprising, since both systems use the same kind of language and try to help users in very similar ways.

This split shows a major weakness in current tech: while systems are great at suggesting things to buy, they aren't very good at answering open-ended questions or helping people who are just 'looking around.' To fix this, we need to stop treating search and recommendations as separate tools. This thesis argues for a single, combined system that can do both through a natural conversation.

Conceptual Framework

Search and recommendation systems have traditionally been developed as distinct paradigms, each addressing different modes of information access. Classical Information Retrieval (IR) systems are query-driven: they assume that users can explicitly articulate their information needs and that the system's task is to retrieve and rank relevant documents from a large

corpus. Recommendation systems, by contrast, are user-driven: they aim to proactively suggest items by modeling user preferences, often without requiring an explicit query [37, 38].

Even though experts treat search and recommendations as two different things, regular users do not see a division. When you type something like 'comfortable sneakers for walking,' you aren't just looking for a definition of shoes, you are asking for a personalized suggestion. Because our real-life needs are often a mix of both, researchers are now working on systems that combine search and recommendation into one single, helpful conversation.

Li and O'Reilly (2017), [25], formalized this intuition by proposing a conceptual framework for joint search and recommendation, arguing that the two tasks should be treated as complementary rather than competing. In their view, search and recommendation represent different points along a continuum of user intent: search emphasizes explicit, short-term goals, while recommendation emphasizes long-term preferences and personalization. A joint system must therefore reason about when to retrieve information, when to recommend items, and how to transition between these behaviors as user intent evolves.

At the core of this framework lies intent understanding. Joint systems must infer whether a user utterance is primarily informational, decisional, or transactional, or whether it combines multiple intents simultaneously. This requirement introduces challenges that are largely absent in traditional IR or recommendation systems when considered in isolation. In particular, intent ambiguity becomes a first-order problem, rather than an edge case, and errors in early intent interpretation can fall into inappropriate system responses.

Motivations for Joint Systems

The motivation for integrating search and recommendation is both practical and theoretical. From a user experience perspective, separating search and recommendation into distinct interfaces or modules imposes unnecessary mental load. Users are forced to adapt their language and interaction strategies depending on the system's capabilities, rather than the system adapting to the user. Studies in interactive IR have shown that users frequently shift between exploration and decision-making within a single session, often without clearly delineating these phases [9, 49].

From a system perspective, joint modeling enables richer use of available signals. Search queries provide high-precision, short-term intent signals, while recommendation histories encode longer-term preferences and behavioral patterns. Treating these signals independently can lead to suboptimal performance, particularly in conversational settings where utterances are short, underspecified, and context-dependent.

The rise of conversational agents has further amplified the need for joint systems. In conversational interfaces, users naturally mix factual questions, comparisons, and preference expressions within a single dialogue turn. For example, a user might ask "what's the difference between hybrid and electric SUVs, and which one would you recommend?" a query that is neither purely search nor purely recommendation. Addressing these requests requires a dual approach: retrieving relevant data and applying reasoning to generate a specific recommendation.

Existing Approaches to Joint Search and Recommendation

Early attempts to combine search and recommendation focused primarily on re-ranking and post-processing. In these systems, search results were first retrieved using standard IR techniques, after which recommendation-style personalization was applied to reorder results based on user profiles or interaction history [46]. While effective in some contexts, these approaches treated recommendation as a secondary layer rather than a fundamentally integrated component.

Subsequent work explored tighter integration strategies. For instance, some systems incorporated recommendation signals directly into the retrieval model, blending relevance and personalization features within a unified ranking function [51]. Others framed recommendation as a special case of retrieval, where user profiles are treated as consistent queries that evolve over time.

Zhang and Chen (2019), [52], provided one of the first comprehensive surveys of joint search and recommendation systems, categorizing existing approaches into several broad families:

- search-based recommendation, where IR techniques are adapted to recommend items;
- recommendation-elevated search, where personalization improves retrieval; and
- fully unified models, which jointly optimize for relevance and personalization.

Despite these advances, the survey highlighted a key limitation: most existing systems operate in non-conversational settings and assume relatively clean separation between user intent types. Mixed or ambiguous intents are often handled heuristically or ignored altogether. Moreover, evaluation protocols typically focus on either retrieval effectiveness or recommendation accuracy, but rarely both.

2.1.3 Conversational Search

Traditional search often assumes the user arrives with a perfectly formed question. In reality, it is not how people work, it is more messy. This is where Conversational Search steps in, it addresses this by shifting the focus from "single-shot" queries to 'back-and-forth' dialogues, allowing users to iteratively sharpen their information needs through natural interaction. You start with a half-formed question, the system asks a follow-up, you clarify, and together you trim down what you are really after. Instead of demanding a perfect initial input, these systems are built to handle the ambiguity, incomplete thoughts, and evolving context inherent in human conversation [33, 34].

Early research in the field fundamentally re-framed the problem. Search is not about finding a document anymore, but about a continuous exchange. This "retrieval-as-interaction" paradigm treats the system as a partner that clarifies intent rather than just a machine that fetches data [19]. When operating over big, open-domain collections like the web, these systems must prioritize relevance estimation, even when the user's true goal is still covered in uncertainty.

The recent surge in neural language models has significantly boosted a system's ability to remember what you said five messages ago and actually understand the casualty in the

way people talk. However, even with these advances, the field faces persistent hurdles. As noted in recent surveys, the "cold start" problem of an early conversational turn remains a major challenge; when context is thin and the user's intent is only implied, the system risks losing the thread entirely [33].

Interestingly, even though search feels more like a conversation now, it's still pretty impersonal. Most current systems prioritize the immediate dialogue over the long-term user's history. While this makes sense for broad, open-world searching, it creates a "fragility" in the system. Because these models lack a deep profile of the user, they are highly sensitive to misinterpretations. If the system misreads a user's goal early on, the entire interaction can quickly veer into irrelevant or misleading territory.

This sensitivity is particularly problematic for mixed-intent systems—those that try to balance both search and recommendations. Understanding how conversational search stumbles when intent is unclear is not just an academic exercise; it is essential for building more robust, reliable AI partners.

2.1.4 Conversational Recommendation

Conversational Recommendation systems aim to guide users toward relevant items through interactive dialogue, typically within a closed-world catalogue such as products, media items, or services. Instead of just finding information, they care about learning what you prefer, narrowing down your choices, and personalizing the results. In these setups, the system does not wait for you to spell out your preferences. It jumps in, asks targeted questions, and keeps nudging you toward something it thinks you will want [53].

Early conversational recommendation frameworks introduced dialogue-driven strategies in which systems actively ask questions to refine user preferences. The System Ask–User Respond idea represents this approach, emphasizing the system's role in steering the interaction toward a recommendation by sequentially reducing uncertainty about user needs [53]. These systems operate on the premise that personalization is essential to their core functionality.

Recent work has further strengthened the role of language in recommendation by developing models that blend natural language and item data. Approaches such as BLAIR demonstrate how large-scale review corpora and item metadata can be leveraged to bridge language and items through contrastive representation learning [22]. These models improve the handling of complex natural language contexts in retrieval and recommendation tasks, but they remain fundamentally recommendation-centric, assuming that the system's primary goal is to surface items rather than information.

A defining characteristic of conversational recommendation systems is that they lean hard on user profiles or historical interaction signals. While this enables strong personalization, prior research has shown that such signals can introduce bias when applied indiscriminately, particularly in early interaction stages where conversational intent is still unclear. Studies on evaluation bias and annotation bias further suggest that recommendation-oriented assumptions can skew system behavior and inflate performance estimates under certain evaluation setups [15].

As a consequence, conversational recommendation systems are robust when user intent aligns with recommendation goals, but they can perform poorly when informational or exploratory intents dominate. This asymmetry becomes problematic in joint conversational

settings, where recommendation mechanisms may overshadow search-oriented needs if intent routing is not handled explicitly.

2.2 Joint Conversational Search and Recommendation

2.2.1 System Perspective

Real-world conversational interactions rarely conform to a single, fixed intent. Users often mix information-seeking and recommendation-oriented behaviors within the same chat or dialog, for example by requesting clarification before committing to a choice, or by asking for recommendations after exploratory search. This observation has motivated early work on Joint Conversational Search and Recommendation (CSR), which aims to unify conversational search and conversational recommendation within a single system [53].

Early CSR systems showed that it is possible for a chatbot to switch between asking clarifying questions, fetching info, and giving suggestions. But there is a catch, most of these systems typically assume what the user wants based on the conversations context. As a result, intent routing is often embedded within heuristic pipelines or end-to-end models, rather than being treated as an explicit decision problem.

Recent surveys highlight that, despite growing interest in conversational systems, the majority of conversational datasets and benchmarks still focus on either search or recommendation in isolation [33]. Even when mixed interactions are considered, evaluation protocols rarely distinguish between failures caused by incorrect intent interpretation and those caused by downstream ranking errors. This makes it difficult to diagnose whether system performance failure comes from misrouting user intent or from limitations of the underlying search or recommendation components.

Another limitation of existing CSR research is the tendency to assume that personalization signals are always a good thing. While user profiles are essential for recommendation tasks, prior work has shown that personalization can be harmful when applied to queries that are informational or exploratory in nature. In joint conversational settings, this creates an imbalance: recommendation-oriented mechanisms tend to be pushier and more confident than search-oriented ones, increasing the risk that recommendation behavior dominates even when search intent is present.

As a consequence, current CSR systems often stumble in early on interactions, where users have not made their intent clear and conversational context is limited. Errors at this stage can propagate through the system, leading to inappropriate responses and bad user experience. Despite this, intent routing itself remains underexplored as a research problem, and is rarely evaluated on its own.

These gaps suggest that progress in CSR requires both suitable datasets and explicit modeling of intent routing under ambiguity. Rather than assuming perfect intent inference, joint conversational systems must be evaluated on their ability to decide when to retrieve information and when to recommend items.

2.2.2 Data and Evaluation Challenges

Shared datasets and agreed-upon evaluation methods have always pushed Information Retrieval (IR) and Recommender Systems forwards. In classical IR, for example, the Cranfield paradigm established the foundation for experimental evaluation by separating system development from evaluation through fixed document collections, topics, and relevance judgments [19]. This setup enables reproducible comparison of systems by controlling for data and ground truth, and has been instrumental in advancing retrieval models over decades.

Conversational systems, however, introduce additional challenges that strain traditional evaluation assumptions. Multi-turn interactions, evolving user intent, and contextual dependencies complicate both data collection and annotation. Collecting good data and labeling it gets way trickier. As a result, most existing conversational datasets are designed around a single dominant task, either conversational search or conversational recommendation, and implicitly assume that intent is fixed or unambiguous throughout an interaction [33].

Conversational search datasets typically deals with open-domain retrieval over large corpora, focusing on relevance judgments conditioned on conversational context. In contrast, conversational recommendation datasets are built around smaller sets of items and rely heavily on user interaction histories and preference signals. While both dataset types have contibuted to valuable research, they are ill-suited for studying joint conversational behavior, as they conflate intent interpretation with downstream ranking performance. When a joint system fails, it is often unclear whether the error comes from the fat that is misunderstood or from limitations of the retrieval or recommendation component itself.

Recent research points out that the way evaluation is designed and setup annotation workflows can significantly bias and skew how we judge a system's performance. Studies on expert finding and related retrieval tasks demonstrate that system-assisted or biased annotations can favor certain model classes and unfair comparison between approaches [15]. Similar concerns have been raised more broadly in the context of modern IR systems, particularly with the integration of large language models, where biases are introduced during data collection, model development, or evaluation can distort conclusions [13]. It is easy to draw the wrong conclusions if you do not look carefully for these traps. That is why building datasets that actively control for bias and keep different parts of the system clearly separated is so important.

Within the context of joint conversational search and recommendation, these issues are particularly strong. Most existing datasets do not provide supervision for intent routing, nor do they separate ground truths for search and recommendation actions. Consequently, evaluation protocols end up rewarding systems that just repeat the most common behaviors, like over-personalizing recommendations, even if the user actually wanted information, not just a suggestion. This limits the ability to study failure modes arising from intent ambiguity and early-stage conversational uncertainty.

CoSRec addresses these limitations by providing a dataset explicitly designed for joint conversational search and recommendation. Unlike prior resources, CoSRec includes conversations spanning pure search, pure recommendation, and mixed-intent interactions, and crucially provides separate ground truths for search-oriented and recommendation-oriented responses. This setup lets you test whether systems can actually figure out what users want before jumping into search or recommendation. It untangles intent recognition from the

ranking step, which means you can judge intent routing on its own.

By sticking to solid experimental controls but also fitting the messy reality of real conversations, CoSRec lets researchers dig into intent routing as its own challenge. You can spot mistakes in routing versus ranking, and see how things like ambiguity, personalization, and context collide in joint systems. In this way, CoSRec is a crucial tool for anyone serious about making joint conversational search and recommendation systems actually reliable and reproducible.

These properties make CoSRec particularly well suited for studying intent routing under ambiguity, and motivate its use as the primary experimental benchmark in the following chapter.

2.3 Semantic ambiguity on intent classification

Intent classification in conversational systems has traditionally been treated as a closed-set multi-class problem, where each user message is supposed to fit into some neat intent label from a fixed list or pre-definded intent taxonomy. However, recent work highlights that real-world user queries often exhibit semantic ambiguity, overlapping intents, or characteristics of open-set intent recognition, particularly when seeking for information and exploratory scenarios. Studies on open and unknown intent recognition show that intents tied to search-oriented queries are often vaguely defined, context-dependent, and harder to separate from similar classes using lexical signals alone [20, 7]. These findings motivate a more nuanced treatment of intent routing in conversational search and recommendation systems. Joint search and recommendation systems face several open challenges:

- Intent ambiguity: User utterances often contain overlapping signals, making it difficult to assign a single dominant intent.
- Semantic asymmetry: Recommendation language tends to be richer and more descriptive, while search language is often short and underspecified, leading to systematic classification biases.
- Personalization bias: User profiles can improve recommendation quality but may distort intent interpretation if applied too early in the pipeline.
- Evaluation complexity: Joint systems must be evaluated along multiple dimensions, including retrieval relevance, recommendation accuracy, calibration, and user satisfaction.

These challenges are exacerbated in conversational settings, where utterances are short, context-dependent, and highly variable. Addressing them requires datasets and experimental frameworks that explicitly model intent overlap and ambiguity rather than treating them as noise.

Positioning of This Work

The limitations identified in prior work motivate the need for datasets and models specifically designed for joint conversational search and recommendation. In particular, there is a lack of benchmark datasets that

- contain both search and recommendation intents,
- allow for mixed-intent annotations, and
- support principled analysis of intent routing decisions.

The CoSRec dataset [8] introduced in this thesis directly addresses these gaps by providing a controlled yet realistic setting in which search and recommendation intents coexist, overlap, and compete. By framing intent routing as a binary, but policy-dependent,decision problem, this work builds on prior conceptual frameworks while enabling systematic empirical analysis of ambiguity, bias, and model behavior in joint CSR systems.

Chapter 3

Problem Description and Setup

3.1 Dataset Overview

The work of this thesis is based on the **CoSRec** dataset, a benchmark specifically designed for *Conversational Search and Recommendation* (CSR). Unlike the traditional datasets that focus solely on either information retrieval or product recommendation, CoSRec models realistic conversational scenarios in which users dynamically alternate between *search-oriented* and *recommendation-oriented* interactions within the same dialogue [8].

In everyday context, users rarely express their needs through a single, fully-specified query. Instead, they engage in multi-turn conversations where their intent evolves over time: they may begin with vague preferences, request general information, refine constraints, and ultimately seek personalized recommendations. CoSRec is designed to capture this behavior by representing user interactions as structured, multi-turn conversations grounded in both document retrieval and product recommendation tasks.

More specifically, the dataset integrates two complementary information sources:

- a large-scale document corpus used to answer informational queries, and
- a product catalog enriched with metadata and user reviews, used to generate recommendations. This dual nature enables the study of systems that must operate across heterogeneous data sources and dynamically adapt to different user needs.

Another defining characteristic of CoSRec is the explicit modeling of **user intent at the utterance level**. Each turn in a conversation is associated with a specific intent, such as information seeking, recommendation requesting, or product-specific inquiry. Importantly, these intents are not fixed throughout the conversation but may shift from one turn to another, reflecting the inherently dynamic nature of human interactions.

The dataset is organized into three subsets, *Raw*, *Crowd*, and *Curated*, which differ in scale and annotation quality.

This design supports both large-scale training and fine-grained evaluation. The global statistics are reported in Table 3.1.

As shown in these tables, CoSRec provides a unique combination of conversational depth, intent diversity, and heterogeneous data sources. This makes it particularly suitable for

Table 3.1: Distribution of conversations, utterances, and relevance judgments across the CoSRec dataset partitions.

Partition	Subset	Conv.	Utt.	Ann. Utt.	Intents	Rel. Judgments (R–P–N)
CoSRecCurated	Search	—	—	—	43	571 – 696 – 1,047
	Rec.	—	—	—	62	5,921 – 4,075 – 5,099
	Details	—	—	—	38	—
	Total	**20**	**150**	**150**	**143**	**6,492 – 4,771 – 6,146**
CoSRecCrowd	Search	—	—	—	1,159	0
	Rec.	—	—	—	1,121	0
	Details	—	—	—	1,988	0
	Total	**291**	**2,329**	**2,329**	**4,268**	**0**
CoSRecRaw	**Total**	**8,938**	**71,656**	**0**	**0**	**0**

Note: Conv. = Conversations; Utt. = Utterances; Ann. Utt. = Annotated Utterances; R–P–N refers to Relevant, Partially Relevant, and Non-Relevant judgments respectively.

evaluating complex pipelines that must jointly handle intent classification, query rewriting, retrieval, and recommendation.

Overall, the CoSRec dataset constitutes a comprehensive benchmark for studying the challenges of conversational systems operating in mixed search and recommendation environments, which aligns closely with the objectives of this thesis.

3.1.1 Dataset Composition

The CoSRec dataset is organized into three complementary subsets, namely *Raw*, *Crowd*, and *Curated*. These subsets differ in terms of scale, annotation quality, and intended usage, and together they provide a comprehensive framework for both large-scale training and rigorous evaluation.

CoSRec-Raw

The **CoSRec-Raw** subset constitutes the largest portion of the dataset, containing 8,938 conversations that are automatically generated. These conversations are constructed using large language models and simulate realistic user-system interactions across multiple turns.

Each conversation in this subset consists of a sequence of user utterances and system responses, capturing the evolving nature of user needs. However, this subset does not include human annotations such as intent labels or relevance judgments.

The primary role of CoSRec-Raw is to provide large-scale conversational data for training data-intensive components, such as neural retrievers, query rewriting models, or ranking

systems. Due to the absence of manual annotations, it is not suitable for reliable evaluation, but it remains essential for learning robust representations of conversational context.

CoSRec-Crowd

The **CoSRec-Crowd** subset contains 291 conversations annotated through crowdsourcing. Compared to the Raw subset, it provides a moderate amount of human-labeled data, enabling the evaluation of intermediate system performance.

In this subset, each conversation is enriched with annotations such as intent labels and relevance judgments. While the annotation quality is generally reliable, it may exhibit some variability due to the nature of crowdsourcing.

As a result, CoSRec-Crowd serves as a bridge between large-scale synthetic data and high-quality expert annotations. It can be used for validation, model selection, and preliminary evaluation.

CoSRec-Curated

The **CoSRec-Curated** subset represents the highest-quality portion of the dataset, consisting of 20 conversations that are manually refined and extensively annotated by experts.

Despite its relatively small size, this subset provides rich annotations that support fine-grained evaluation across multiple dimensions. Each utterance is labeled with its corresponding intent, and both document retrieval and recommendation tasks are associated with detailed relevance judgments.

A distinctive feature of this subset is the inclusion of multiple user profiles for each conversation. These profiles are derived from user preferences and historical interactions, allowing the evaluation of personalized recommendation systems. Consequently, the same conversation may yield different relevant items depending on the user profile considered.

In addition, the curated subset includes response quality annotations, such as fluency, coherence, and informativeness, enabling evaluation beyond traditional retrieval metrics.

Comparison of Subsets

The three subsets are designed to complement each other, forming a hierarchical structure that balances scale and annotation quality.

- **CoSRec-Raw**: large-scale, unannotated data for training.
- **CoSRec-Crowd**: moderately sized, crowdsourced annotations for validation and intermediate evaluation.
- **CoSRec-Curated**: small-scale, expert-annotated data for rigorous and reliable evaluation.

This design reflects a common trade-off in conversational datasets between quantity and quality. By combining these subsets, CoSRec enables the development of systems that can leverage large amounts of data while still being evaluated under controlled and high-quality conditions.

Implications for This Work

In this thesis, the different subsets are used according to their respective strengths. The CoSRec-Raw subset is primarily used for training components that require large amounts of conversational data, while the CoSRec-Curated subset serves as the main benchmark for evaluation due to its high-quality annotations. When applicable, the CoSRec-Crowd subset is used for intermediate validation and analysis.

This separation ensures that the evaluation results are reliable and aligned with human judgment, while still allowing the models to benefit from large-scale training data.

3.1.2 Structure of a Conversation

The fundamental unit of the CoSRec dataset is a **multi-turn conversation**, which models the interaction between a user and a conversational system. Each conversation consists of a sequence of turns, where each turn is composed of a *user utterance* followed by a *system response* [33].

Unlike traditional retrieval datasets, where queries are treated as independent and self-contained, conversations in CoSRec are inherently **context-dependent**. Each user utterance must be interpreted in relation to the preceding dialogue, as users often omit information, refer to previous entities, or refine their requests incrementally.

Turn-Level Structure

Formally, a conversation can be represented as an ordered sequence of turns:

$$C = \{(u_1, s_1), (u_2, s_2), \ldots, (u_T, s_T)\} \tag{3.1}$$

where u_t denotes the user utterance at turn t, and s_t denotes the corresponding system response. The total number of turns T varies across conversations, reflecting different levels of interaction complexity.

Each user utterance expresses a specific information need, which may depend on previous turns. Consequently, the system must consider the full conversational history $\{u_1, \ldots, u_{t-1}\}$ when generating an appropriate response.

Illustrative Example

To better understand the structure of a conversation, Table 3.2 presents a representative example extracted from the dataset.

As shown in Table 3.2, the conversation evolves over multiple turns, with the user progressively refining their request. The interaction may begin with a broad or underspecified query, such as a request for product recommendations, followed by more specific questions about product attributes, ingredients, or usage.

For instance, a user might initially ask for recommendations for skincare products, then inquire about the benefits of a particular ingredient, and finally request details about a specific item. This progression illustrates how user intent is not static, but rather evolves dynamically throughout the conversation.

Utt.		Utterance Text
1	**U:**	Hi there! I'm looking for a new pillow. I've been sleeping terribly lately, and I think it's time for an upgrade.
	S:	I understand! [...] To start, what kind of pillow are you looking for? Do you prefer memory foam, down alternative, or something else?
2	**U:**	I'm open to different materials, but I want something that will keep me cool at night. I tend to sleep hot.
	S:	That's a great point. Many people find that memory foam pillows can trap heat. Have you considered down alternative pillows? They often have a good breathability, and they are often more affordable than down.
3	**U:**	Down alternative sounds good. I'm not sure what brands are good though. Any suggestions?
	S:	There are many great brands out there! For down alternative pillows, some of the top contenders are Beckham Hotel Collection, [...].
4	**U:**	Okay, Beckham Hotel Collection... I think I've heard of that brand. What makes them good?
	S:	Beckham Hotel Collection is known for making high-quality pillows [...]. They also have a gel cooling feature, which might be perfect for you since you mentioned you tend to sleep hot.
5	**U:**	Gel cooling? That sounds interesting! How does it actually work?
	S:	The gel is infused within the pillow's filling, which helps to absorb and dissipate heat. [...].
6	[...]	

Table 3.2: Example of conversation from the CoSRec dataset.

Context Dependency

A key characteristic of conversational data in CoSRec is the strong dependency on context. Many user utterances are *underspecified* when considered in isolation and require information from previous turns to be fully understood.

For example, a follow-up question such as *"Is it suitable for sensitive skin?"* does not explicitly mention the product under discussion. The system must look back from earlier turns in the conversation. This adds or introduces additional complexity compared to traditional retrieval settings and necessitates the use of techniques such as context modeling and query rewriting.

Dynamic Intent Evolution

Another important aspect of conversation structure is the presence of **dynamic intent evolution**. Within a single dialogue, the user may alternate between different types of information needs, such as general information seeking, recommendation requests, and product-specific inquiries.

This behavior reflects realistic user interactions, where exploration and decision-making occur iteratively. As a result, the system must continuously adapt its behavior based on the current conversational context and inferred user intent.

Implications for System Design

The multi-turn and context-dependent nature of conversations in CoSRec introduces several challenges for system design. First, models must effectively capture long-range dependencies across turns. Second, they must resolve implicit references and incomplete queries. Third, they must adapt to changing user intents within the same conversation.

These challenges motivate the need for modular architectures that explicitly model conversational context and dynamically adjust system components, such as intent classification, query rewriting, and retrieval or recommendation strategies.

The example conversation shown in Table 3.2 is drawn from the CoSRec dataset, specifically from its curated subset, which contains high-quality, manually refined interactions [33].

3.1.3 Intent Taxonomy

A central property of the CoSRec dataset is the explicit annotation of **user intent at the utterance level**. In contrast to conventional retrieval datasets, where each query is typically associated with a single dominant objective, CoSRec models conversations in which the user may shift between different goals across turns, and in some cases may even express more than one goal within the same utterance. This makes intent annotation a core component of the dataset design, as it provides the semantic layer needed to understand how conversational search and recommendation interact within a unified setting [8].

In CoSRec, each utterance can be annotated with zero, one, or more labels from three intent categories:

- **Search**
- **Recommendation**
- **Product Detail**

These three labels capture distinct but closely related forms of user need. Taken together, they define the action space that a CSR system must interpret before deciding how to respond.

Search Intent

The **search** intent corresponds to utterances in which the user asks for general, explanatory, or factual information about a topic related to the product under discussion. These requests are not primarily about selecting a specific item, but about obtaining knowledge that helps the user better understand the domain, compare alternatives, or clarify product-related concepts.

Typical search utterances are open-ended in nature and often resemble informational queries that would traditionally be answered through document retrieval. For example, a user may ask:

> "What are some common ingredients used in these products?"

or

> "Can you tell me more about how lavender helps with skincare?"

In such cases, the answer is expected to come from an external information source, such as the document corpus associated with the dataset, rather than directly from the product catalogue itself. Search intent therefore represents the *open-world* side of the CSR setting, where the system must retrieve or generate factual information that may go beyond any single item description.

Recommendation Intent

The **recommendation** intent refers to utterances in which the user asks the system to suggest one or more products that match their needs, preferences, or constraints. Unlike search, which is information-centered, recommendation is decision-centered: the goal is not merely to explain the space of options, but to guide the user toward suitable items.

Recommendation utterances often contain subjective preferences, functional needs, or desired product characteristics. For example, a user may say:

> "I'm looking to buy a product that will help me remove makeup and take care of my skin."

or

> "Can you recommend some products that include lavender?"

These requests are naturally handled within a *closed-world* setting, where the system must identify relevant items from the available product catalogue. Recommendation intent is also the most directly connected to personalization, since different users may prefer different products even when they express similar needs.

Product Detail Intent

The **product detail** intent corresponds to utterances in which the user asks for specific information about a particular item that has already been introduced in the conversation. This intent differs from general search because the target of the question is not an open topic, but a known product under discussion.

Typical examples include:

> "Is this product suitable for dry and sensitive skin?"

or

> "Can you tell me the size, brand, or material of this item?"

The key distinction between *search* and *product detail* lies in the expected source of the answer. A product-detail question can usually be answered by inspecting the metadata or description of the referenced item, while a search question requires broader external knowledge. For this reason, product detail forms a separate intent category in CoSRec, even though it remains closely related to both search and recommendation.

Intent Overlap and Multi-Label Nature

One of the most important characteristics of CoSRec is that intent labels are **not mutually exclusive**. A single utterance may contain multiple intents, such as shown in 3.3, reflecting the fact that real conversational behavior is often mixed and incremental rather than neatly separable.

For example, a user might ask for a recommendation while simultaneously including an informational component:

> "Can you recommend a product for sensitive skin, and what ingredients should I look for?"

Such an utterance combines a recommendation request with an informational need. This

User Need	Purchase a new bed pillow.
User Utt.	I don't have a specific material in mind, but I do want something that's machine washable and dryable. And I'd prefer a standard size.
Intent #1	*Span of Text*: I don't have a specific material in mind *Search Intent*: materials used for bed pillows
Intent #2	*Span of Text*: but I do want something that's machine washable and dryable. And I'd prefer a standard size. *Recommendation Intent*: standard size bed pillow which is machine washable and machine dryable

Table 3.3: Example of multiple intents in single user utterance.

is precisely the type of ambiguity that makes CoSRec particularly valuable for studying intent

routing. Instead of assuming that each turn belongs to one isolated category, the dataset allows the analysis of overlapping and evolving goals, which more closely reflects natural dialogue.

Distribution of Intent Labels

The dataset includes intent annotations for both the Crowd and Curated subsets. These distributions provide an important quantitative view of the dataset, showing that the three intent types are all substantially represented.

The intent statistics reveal two important properties of the dataset. First, the annotation space is not dominated by a single intent type, meaning that CoSRec supports a balanced study of multiple conversational behaviors. Second, the presence of all three labels across both subsets confirms that conversations are structurally heterogeneous: users do not interact only to retrieve information or only to receive recommendations, but instead move between these needs throughout the dialogue.

Why the Intent Taxonomy Matters

This taxonomy is essential for the goals of this thesis. Since the broader CSR pipeline must decide how to interpret and process each user utterance, intent labels provide the first level of semantic organization over the conversations. They make it possible to study not only whether a system retrieves relevant documents or recommends suitable items, but also whether it correctly identifies the kind of need expressed by the user in the first place.

More broadly, the CoSRec intent taxonomy formalizes the central challenge of joint conversational search and recommendation: conversational utterances often lie at the boundary between information access and decision support. By explicitly annotating search, recommendation, and product-detail intents, the dataset offers a structured way to analyze this boundary and to evaluate systems operating under mixed and evolving conversational goals.

3.1.4 Data Sources

A defining characteristic of the CoSRec dataset is the integration of **heterogeneous data sources**, which enables the modeling of both conversational search and conversational recommendation within a unified framework. In particular, the dataset combines two complementary resources:

- a large-scale document corpus and
- a product catalog enriched with metadata and user-generated content [8].

Document Corpus

The document corpus is used to support **search-oriented interactions**, where the user seeks general or explanatory information. In these cases, the system is expected to retrieve relevant passages that provide factual or descriptive answers to the user's query.

In CoSRec, this component is based on a large-scale web document collection, such as MS MARCO, which contains millions of passages covering a wide range of topics. This corpus

enables the system to answer open-ended questions that are not tied to a specific product, such as explanations of ingredients, usage guidelines, or comparisons between different product categories.

The use of a document corpus introduces an *open-world* setting, where the system must retrieve information from a broad and diverse knowledge base. This is fundamentally different from recommendation tasks, where the candidate space is restricted to a predefined set of items.

Product Catalog

The second data source is a **product catalog**, which supports recommendation and product-specific queries. This catalog is derived from e-commerce data and includes detailed information about items, such as titles, descriptions, categories, and Amazon product reviews.

The product catalog is primarily used to handle:

- **Recommendation intents**, where the system must suggest relevant products based on user preferences.
- **Product detail intents**, where the system must provide specific information about a given item.

Unlike the document corpus, the product catalog defines a *closed-world* setting, where the system operates over a finite and structured set of items. This enables ranking-based approaches and personalized recommendation strategies, as relevance can be defined with respect to user preferences.

Complementarity of Data Sources

While MS-MARCO and the Amazon catalogue are structurally independent corpora, they functionally complement each other within the CoSRec framework. This allows the system to support a unified user journey that transitions from open-world information seeking (Search) to closed-catalogue selection (Recommendation). This dual-source environment introduces complexity for the Intent Router, which must dynamically bridge the gap between these heterogeneous backends.

Implications for System Design

The presence of multiple data sources has important implications for the design of conversational systems. First, it requires mechanisms to route user queries to the appropriate component, depending on whether the task involves document retrieval or product recommendation. Second, it motivates the use of unified architectures that can integrate signals from both sources.

In the context of this thesis, this dual-source setting directly motivates the proposed pipeline, where intent classification determines whether a query should be handled by a retrieval module or a recommendation module, and query rewriting ensures that context-dependent utterances can be effectively processed by both.

Overall, the integration of heterogeneous data sources in CoSRec provides a realistic and challenging testbed for studying conversational systems that must operate across both information access and recommendation scenarios.

3.1.5 Annotation Scheme

The CoSRec dataset provides a rich annotation framework that enables fine-grained evaluation of conversational systems across multiple dimensions. Unlike traditional datasets that focus solely on retrieval relevance, CoSRec includes annotations for **intent classification**, **relevance judgments**, and **response quality**, reflecting the complexity of conversational search and recommendation scenarios [8].

These annotations are primarily available in the Crowd and Curated subsets, with the Curated subset offering the highest level of annotation quality and consistency.

Intent Annotations

Each user utterance is annotated with one or more intent labels, as described in Section 3.3. These labels indicate whether the user is seeking general information (search), requesting product suggestions (recommendation), or inquiring about a specific item (product detail).

Intent annotations form the basis for evaluating intent classification models. Since multiple intents may co-occur within a single utterance, the annotation scheme allows for multi-label assignments, reflecting the complexity of natural conversational interactions.

Document Relevance

For search-oriented queries, relevance is assessed at the level of retrieved documents or passages. The dataset adopts a graded relevance scale, where each candidate document is assigned a relevance level indicating its usefulness in addressing the user's query.

Typically, the scale follows three levels:

- 0: Not relevant
- 1: Partially relevant
- 2: Highly relevant

This graded scheme enables the use of ranking-based evaluation metrics such as nDCG, which reward systems for placing highly relevant documents at higher ranks.

3.1.6 Dataset Statistics and Analysis

To better understand the characteristics of the CoSRec dataset, this section provides a quantitative analysis of its main components, including conversation structure, intent distribution, and relevance judgments. These statistics offer insights into the complexity of the dataset and highlight the challenges faced by conversational systems operating in this setting [8].

Conversation-Level Statistics

The dataset comprises a total of 9,249 conversations, distributed across the Raw, Crowd, and Curated subsets.

As shown above in Table 3.1, the majority of conversations belong to the Raw subset, which provides large-scale training data, while the Curated subset contains a small number of high-quality annotated conversations.

Although the number of conversations in the Curated subset is limited, each conversation is carefully constructed and annotated, making it suitable for fine-grained evaluation. In contrast, the Raw subset emphasizes scale over annotation depth, reflecting a common trade-off in conversational datasets.

In terms of length, conversations consist of multiple turns, with each turn representing a user-system interaction. This multi-turn structure increases the complexity of the task, as models must capture dependencies across the entire conversation rather than treating each query independently.

Intent Distribution

The distribution of intent labels provides important insights into how users interact within the dataset. As shown in Table 3.4, all three intent types—search, recommendation, and product detail, are well represented.

Table 3.4: Statistics of labelled intents by data source (S = Search, R = Recommendation, P = Product Details)

Intent Type	**S**	**R**	**P**	**Total**
Curated	43	62	38	143
Crowd	1159	1121	1988	4268

The statistics of the table above indicate that conversations are not dominated by a single type of intent. Instead, users frequently alternate between different intents, reflecting real-word behavior in which information gathering and decision-making are merged.

For example, a user may begin with a recommendation request, then ask informational questions to better understand product features, and finally request details about a specific item. This pattern confirms that conversational interactions in CoSRec are inherently heterogeneous and dynamic.

Relevance Judgment Distribution

Relevance judgments provide another important dimension for understanding better the dataset. Tables 3.4 and 3.5 summarize the distribution of relevance labels across annotated items.

Table 3.5: Statistics of relevance judgments by intent type

Relevance	Search	Recommendation	Total
Highly Relevant	571	5921	6492
Partially Relevant	696	4075	4771
Not Relevant	1047	5099	6146

These results show that the dataset contains a diverse range of relevance levels, including highly relevant, partially relevant, and irrelevant items. This distribution is essential for evaluating ranking models, as it enables the use of graded relevance metrics such as nDCG.

Furthermore, the presence of both relevant and non-relevant candidates ensures that the evaluation setting is realistic and non-trivial. Systems must not only retrieve relevant items but also rank them appropriately among less relevant alternatives.

Personalization and User Profiles

An important aspect of the dataset statistics is the role of user profiles in the evaluation of recommendation tasks. In the Curated subset, each conversation is associated with multiple user profiles, which introduce variability in relevance judgments.

This means that the same conversation may lead to different sets of relevant products depending on the user profile considered. As a result, the dataset supports personalized evaluation, where system performance must be assessed relative to individual user preferences rather than a single global ground truth.

This property increases the complexity of the task, as systems must not only identify relevant items but also tailor their recommendations to specific users.

Implications for System Evaluation

The statistical properties of CoSRec have important implications for evaluation. First, the multi-turn structure of conversations requires models to capture contextual dependencies, making simple query-based approaches insufficient. Second, the balanced distribution of intent types necessitates systems that can handle multiple tasks within a unified framework.

Third, the graded relevance annotations enable fine-grained evaluation of ranking performance, while the presence of personalization introduces an additional layer of complexity. Finally, the relatively small size of the Curated subset implies that evaluation results must be interpreted carefully, as they may be sensitive to variance.

Summary and Discussion

Overall, the statistics of the CoSRec dataset confirm that it provides a challenging and realistic benchmark for conversational systems. The combination of multi-turn interactions, dynamic intent switching, heterogeneous data sources, and personalized relevance judgments creates a complex evaluation environment that closely mirrors real-world scenarios.

These characteristics justify the need for modular and adaptive approaches, such as the pipeline proposed in this thesis, which explicitly addresses intent classification, query rewriting, and the integration of retrieval and recommendation components.

3.2 Problem Definition and Setup

This thesis addresses the problem of *intent routing* in joint Conversational Search and Recommendation (CSR) systems. At each turn of a conversation, the system must determine how to interpret a user utterance, namely whether it should be handled as an information-seeking request or as a personalized recommendation request.

This decision precedes all downstream components, such as retrieval, ranking, and response generation, and therefore plays a critical role in shaping system behavior and overall response quality. Errors at this stage propagate through the pipeline and may lead to fundamentally inappropriate responses.

Formally, let u_t denote a user utterance at turn t, and let c_t denote the corresponding conversational context, consisting of previous user and system turns. The intent routing task consists of learning a function:

$$f(u_t, c_t) \rightarrow y_t \tag{3.2}$$

where $y_t \in \{\text{search}, \text{recommendation}\}$ represents the predicted intent class associated with the utterance.

The task is studied using the CoSRec dataset, which contains 9,249 conversations spanning multiple turns [8]. Since each conversation is composed of a sequence of utterances, the total number of routing instances is substantially larger than the number of conversations. Each utterance constitutes a separate classification decision, making intent routing a turn-level prediction task. Furthermore, due to the multi-turn nature of the data, many utterances are context-dependent and cannot be interpreted in isolation.

In this work, intent routing is explicitly treated as a standalone classification problem. This formulation abstracts away from downstream processes such as retrieval or generation, allowing routing errors to be analyzed independently. By isolating this stage, it becomes possible to identify when and why routing decisions fail, without confounding effects from later components in the pipeline.

Intent routing in CSR is inherently challenging due to the ambiguity of user language. Utterances are often short, underspecified, and lack explicit signals that clearly distinguish between informational and recommendation-oriented requests. Moreover, similar surface forms may correspond to different intents depending on conversational context. As a result, routing decisions must often be made under uncertainty.

The goal of this chapter is to formally define the intent routing task, describe the models used to address it, and outline the experimental setup adopted in this work.

3.2.1 The Intent Routing Task

Intent routing in CSR systems requires distinguishing between user utterances that seek information and those that seek recommendations. In this thesis, *search intent* refers to

utterances whose primary goal is to obtain factual, explanatory, or comparative information from an open-domain corpus. In contrast, *recommendation intent* refers to utterances whose goal is to receive personalized suggestions drawn from a fixed item catalog.

Although this distinction is conceptually clear, it is often difficult to infer in practice. User utterances may contain overlapping signals, such as references to products within informational questions or expressions of preferences framed as factual queries. For example, a user asking about the differences between two products may implicitly precede a recommendation request, while a recommendation-oriented utterance may require factual clarification before it can be satisfied.

A key challenge arises from the asymmetry between search and recommendation. Conversational recommendation systems often rely on personalization signals, such as user profiles or historical preferences, whereas conversational search primarily depends on the linguistic content of the current utterance. When these signals are combined prematurely, recommendation bias may dominate the routing decision, causing the system to favor recommendation behavior even when informational intent is present.

To address this issue, intent routing is modeled as a supervised classification task, where each utterance is assigned a single primary intent label. Although real-world interactions may involve mixed or evolving intents, this formulation provides a controlled setting for analyzing routing behavior.

For example, the following sequence of user utterances within a conversation:

- u_1: "Can you recommend a good moisturizer for dry skin?" → **Recommendation**
- u_2: "What ingredients should I look for in a moisturizer?" → **Search**
- u_3: "Is this product suitable for sensitive skin?" → **Recommendation**

This example illustrates two important properties of the task. First, user intent may shift across turns, requiring the system to adapt its routing decisions dynamically. Second, some utterances are ambiguous when considered in isolation and require conversational context for correct interpretation [8].

By isolating intent routing from downstream components, this work focuses on understanding how linguistic cues, conversational context, and personalization signals influence routing behavior, particularly under ambiguity.

3.2.2 Models for Intent Routing

To analyze intent routing under different modeling assumptions, this work evaluates a range of models with increasing representational capacity. These models reflect common approaches to text classification in information retrieval and conversational systems.

Linear Models. As a baseline, linear classifiers are trained using sparse lexical features derived from user utterances. In particular, logistic regression is employed as a simple yet effective model. Despite their simplicity, linear models can capture strong lexical signals and often perform competitively when discriminative keywords are present.

Tree-Based Models. To explore non-linear decision boundaries, tree-based models such as random forests are considered. These models can capture interactions between features that are not accessible to linear classifiers. However, when applied to high-dimensional sparse text representations, they often struggle to generalize effectively.

Neural Encoder Models. To capture richer semantic representations, transformer-based neural models are used to encode user utterances into dense contextual representations. These models go beyond surface-level lexical matching and are better suited for handling paraphrases, implicit intent signals, and underspecified language.

In particular, this work focuses on **DeBERTa**-based encoders, which employ disentangled attention mechanisms to separately model content and positional information. This design improves representation quality, especially for short and ambiguous utterances.

For each neural configuration, the encoder produces a fixed-size representation of the input, which is then passed to a lightweight classification layer to predict the intent label. In extended settings, additional inputs such as conversational context and user profile information are incorporated to study their effect on routing performance.

3.2.3 Training Objective and Loss Functions

Intent routing is formulated as a supervised classification task, where the objective is to correctly predict the appropriate system action for a given user utterance. Models are trained to minimize a loss function that penalizes incorrect routing decisions, with particular attention to class imbalance and the inherent ambiguity of conversational language.

Binary Classification Objective. In the primary setting considered in this thesis, intent routing is modeled as a binary classification problem, distinguishing between search-oriented and recommendation-oriented intents. Let $y \in \{0, 1\}$ denote the ground-truth intent label for an utterance, and let $\hat{y} \in [0, 1]$ denote the model's predicted probability of the positive class.

Training is performed using the binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}} = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right]. \tag{3.3}$$

Binary cross-entropy penalizes confident misclassification and provides a smooth optimization objective for probabilistic classifiers. It is widely used in binary text classification tasks and serves as a natural choice for intent routing.

Class Imbalance and Weighted Loss. The distribution of intent labels in conversational data is often imbalanced. In the CoSRec dataset, recommendation-oriented utterances tend to appear more frequently than search-oriented ones, reflecting the prevalence of product-focused interactions [8].

To mitigate bias toward the majority class, a weighted variant of the binary cross-entropy loss is employed:

$$\mathcal{L}_{\text{WBCE}} = -\left[w_+ \, y \log(\hat{y}) + w_- \, (1 - y) \log(1 - \hat{y})\right], \tag{3.4}$$

where w_+ and w_- are class-specific weights inversely proportional to class frequencies. This weighting increases the penalty for misclassifying underrepresented intents and encourages the model to attend to weaker search signals that might otherwise be overshadowed by recommendation bias.

Optimization and Fine-Tuning Strategy. For transformer-based models, fine-tuning is performed using gradient-based optimization over the entire network. To stabilize training and preserve useful pretrained representations, layer-wise learning rate decay (LLRD) is applied. Under this strategy, higher learning rates are assigned to task-specific layers, while progressively lower learning rates are used for lower encoder layers. This approach has been shown to improve convergence and generalization when adapting pretrained language models to downstream classification tasks.

Scope of the Training Objective. The training objective is deliberately restricted to routing accuracy, without directly optimizing downstream retrieval or recommendation performance. This design choice allows intent routing to be evaluated as an independent component of the overall system.

By isolating this stage, it becomes possible to analyze routing behavior without confounding effects from later components in the pipeline. In particular, this setup enables a clearer understanding of how well models capture user intent, independently of the effectiveness of retrieval or recommendation modules.

This is particularly important in CSR settings, where misclassification may lead to fundamentally different system behaviors.

Chapter 4

Methodology

This chapter presents the methodological framework adopted to study and analyze intent routing whithin joint Conversational Search and Recommendation (CSR) systems. Rather than treating intent as a vague concept we formulate the task as a supervised classification problem at the utterance level, where the goal is to determine whether a user request should be handled as a search or recommendation action. In addition to this primary binary formulation, a multi-label setting is also considered to capture overlapping and ambiguous intent signals.

The analysis is done using the CoSRec dataset, where models are trained on large-scale conversational data and then evaluated on the curated test set, which provides high-quality annotations for reliable assessment that by using this expert-verified data rather than "noisy" synthetic or Crowdsourced labels, we can uncover critical architectural issues. Each user utterance is treated as an individual prediction instance, reflecting the turn-level of intent routing nature.

In this chapter we explain how we built a system to help an AI decide what a user actually wants during a conversation. In these systems, a user is usually doing one of two things: searching for a specific fact or looking for a personal recommendation. It then presents the range of machine learning approaches evaluated, including linear models such as logistic regression, tree-based methods such as random forests, and transformer-based neural architectures, with a particular focus on DeBERTa encoders that can understand the deep meaning and emotion behind language.

Finally, the chapter describes the training objectives and optimization strategies used to learn these intent routing models, including binary and multi-label cross-entropy losses, class imbalance handling, and fine-tuning techniques such as layer-wise learning rate decay. Together, these components define a systematic framework for analyzing intent routing behavior under different modeling assumptions and input configurations. evaluated on the curated subset, which contains high-quality manually verified annotations.

4.1 Label Distribution

CoSRec is annotated with three intent labels: *search*, *recommendation*, and *product_details*. An utterance may be associated with multiple intents, reflecting so the inherently mixed and

evolving nature of user behavior in conversational settings.

The statistics reported in this section are computed over the **curated test set** of CoSRec, which provides high-quality manually verified annotations. All analyses are performed at the **utterance level**, where each of the utterance are treated as an independent unit that may be also related with one or more intent labels.

Table 4.1: Absolute and Relative Frequency of Target Labels in the curated test set (utterance-level).

Label	**Absolute Count**	**Relative Frequency (%)**
search	917	56.19
recommendation	887	54.35
product_details	831	50.92

In detail, Table 4.1 presents the distribution of target labels within the annotated dataset, using both absolute counts, that is the total count of messages/turns, and relative theirfrequencies. The relative frequency represents the proportion of the total annotated utterances ($N \approx 1,632$) that contain a specific intent. Notably, the frequencies for Search (56.19%), Recommendation (54.35%), and Product Details (50.92%) sum to approximately 161.46%. This total exceeds 100% because the study employs a multi-label framework, where a single user utterance, such as *"I'm looking for a coffee maker (Search), can you recommend one under $50? (Recommendation)*" can be assigned multiple categories simultaneously.

This near uniform distribution across classes is a desirable property of the dataset, this way it makes sure that no single intent dominates the training process, so preventing the models from developing a majority class bias. Furthermore, the high intent density reflects the complex, overlapping nature of real-world human dialogue, where users often combine different requests within a single conversational turn. To better understand the relationships between these intent categories, we analyze their pairwise co-occurrence within the same utterances. Two labels are said to co-occur if they are both assigned to a single utterance.

Figure 4.1 visualizes the frequency of such co-occurrences across all annotated utterances in the curated test set.

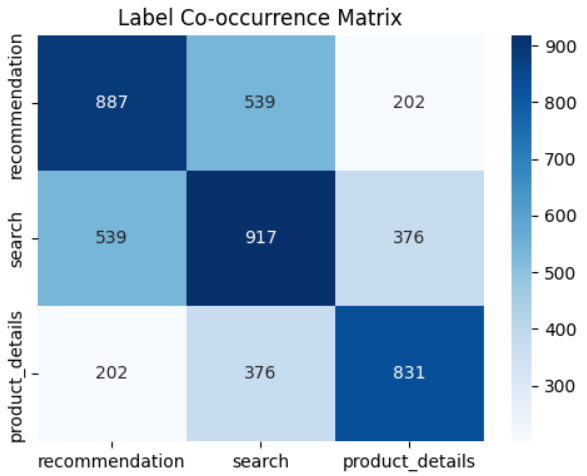


Figure 4.1: Heatmap visualizing the frequency of pairwise label co-occurrence in user utterances (curated test set).

The diagonal cells correspond to the total number of utterances associated with each intent, while the off-diagonal cells represent the number of utterances in which two intents are assigned simultaneously.

A clear asymmetry appears from the co-occurrence structure. The *search* intent reveals substantial overlap with both *recommendation* and *product_details* (539 and 376 co-occurrences), indicating that informational queries frequently co-occur with downstream decision-making or specification requests, reinforcing so the role of Search intent as a transitional bridge between a user's initial information-seeking and the final action of choosing or purchasing an item. In contrast, the co-occurrence between *recommendation* and *product_details* is comparatively limited (202 instances), suggesting that these intents remain more distinct.

This pattern highlights that ambiguity in conversational utterances is not uniformly distributed across labels, but is primarily driven by search-oriented language. Search acts as a transitional intent, bridging exploratory information needs with more concrete actions such as product selection or evaluation.

These observations have important implications for intent routing. In particular, they suggest that distinguishing search from other intents constitutes the main challenge, while recommendation and *product_details* are more easily separable.

4.1.1 Utterance Sparsity and Structure

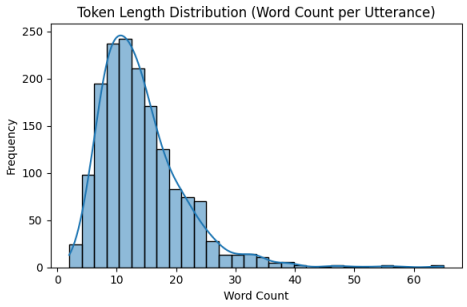
To get a deeper understanding we looked at short-text classification that is frequently hindered by utterance sparsity, where the limited number of tokens provides few features for pattern recognition. In such regimes, prior studies have demonstrated that linear classifiers often outperform complex non-linear models [33]. Non-linear architectures may overfit to noise or struggle to find meaningful interactions within a restricted feature space. Consequently, the transparency and stability of linear weighting prove advantageous when processing brief, information-sparse user inputs.

CoSRec utterances are characteristically short and telegraphic. As shown in Figure 4.2, the distribution of utterance length is heavily skewed toward very short inputs, with a strong mode at 4–5 tokens and a median length of 6 tokens. Fewer than 3% of utterances exceed 20 tokens. This confirms that the dataset exhibits a high degree of lexical sparsity, where individual tokens carry disproportionately high predictive weight.

This sparsity has important modeling implications. In such cases, linear classifiers operating on TF–IDF representations can perform well, as they assign global weights to highly discriminative tokens. This observation helps explain the strong initial performance of Logistic Regression and linear SVM models observed in earlier experiments.

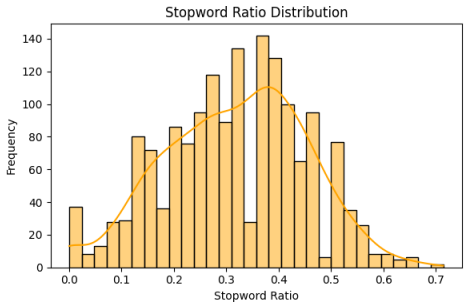
Figure 4.3 complements this analysis by showing the distribution of stop-word ratios. The presence of a substantial proportion of functional words (e.g., interrogatives and prepositions) indicates that intent expression is not driven solely by topical content words. Instead, syntactic and discourse-level cues play an important role in distinguishing between intents, such as differentiating informational queries from search commands. This motivates the use of contextual transformer models, such as DeBERTa, which are better equipped to model functional word usage and subtle syntactic patterns.

Together, these observations characterize CoSRec as a sparse, short-text dataset where linear models are effective for clear cases, while contextual representations are necessary to resolve ambiguity in intent expression.



..

Figure 4.2: Distribution of utterance length (token count).



..

Figure 4.3: Distribution of the ratio of stop words to total tokens.

Key Predictive Features

Analyzing the top features using χ^2 (Chi-Squared) and Mutual Information (MI). χ^2 highlights tokens with strong linear association to a single class, which are the signals leveraged by Logistic Regression.

Table 4.2: Top 5 Features per Label, ranked by χ^2 and Mutual Information (MI).

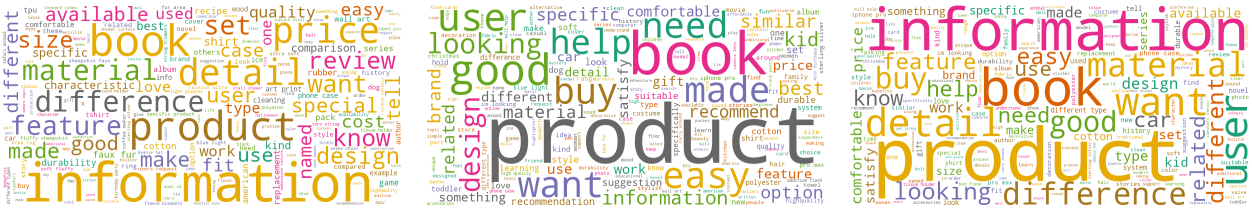
Label	**Top 5 χ^2 (Linear Indicators)**	**Top 5 MI (High Frequency)**
recommendation	*information, that, for, product, products*	*and, the, for, of, about*
search	*by, looking, sandals, price, rainbow*	*and, the, for, of, to*
product_details	*that,* ***details****, information, for, looking*	*and, the, for, of, to*

The distinct, high-χ^2 features (e.g., "details" for *product_details* or "reviews" for *recommendation*) demonstrate that the intents have a lineraly separable nature . These features allow Logistic Regression to assign a high global weight to them, making predictions robust even in a sparse feature space. In contrast, the Random Forest model's inability to split effectively on these critical features confirms the hypothesis for its underperformance.

Word Cloud Analysis for Intent Classes

To further understand the semantic composition of the dataset, generation of **word clouds** for each intent label was provided. Each word cloud visually represents the most frequent terms within the utterances belonging to a specific class. Larger words indicate higher term frequency after removing stop words and applying standard preprocessing (lowercasing, punctuation stripping, lemmatization).

Figure 4.4 presents the resulting word clouds for the three intent labels: *product_details*, *recommendation*, and *search*.



(a) *product_details* (b) *recommendation* (c) *search*

Figure 4.4: Word cloud visualization for each intent label. Larger words denote higher token frequency.

The word clouds confirm clear thematic differences between the three intents:

- **Product Details:** The most prominent terms include *information*, *product*, *material*, *difference*, *size*, and *feature*. These reflect user utterances that seek factual information about an item's specifications or comparison between alternatives (e.g., "What is the difference between cotton and polyester shirts?"). The frequent co-occurrence of terms

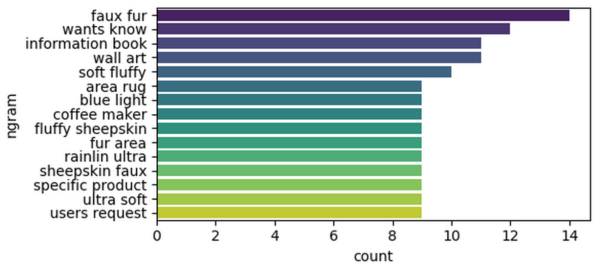
like *detail*, *design*, and *made* suggests a descriptive and attribute-focused linguistic pattern.

- **Recommendation:** The dominant terms such as *product*, *buy*, *help*, *good*, and *recommend* illustrate users expressing purchase intent or seeking suggestions ("Which book should I buy?"). The prevalence of adjectives (*good*, *best*, *easy*) and verbs (*buy*, *need*) indicates an evaluative and decision-oriented discourse structure.
- **Search:** The vocabulary of the search intent shows an overlap with recommendation and product detail, but with stronger emphasis on *looking*, *find*, and *specific*. This class tends to reflect exploratory behaviors, where the user queries for general availability or attempts to locate items ("find a blue book", "search for similar design").

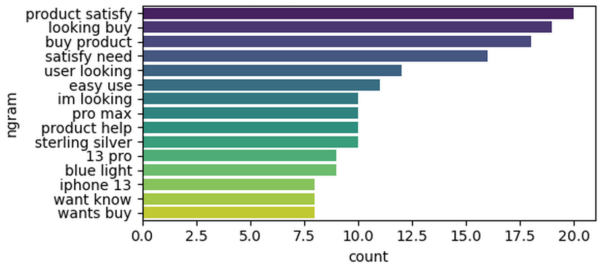
These semantic distributions confirm that the dataset captures distinct functional patterns between the three intent types. While *product_details* queries focus on informational completeness, *recommendation* queries highlight subjective preferences, and *search* queries reflect navigational or lookup behavior. However, frequent overlaps between the most common tokens (*product*, *information*, *book*) also reveal potential sources of misclassification, particularly between *recommendation* and *search* intents.

Frequent N-gram Analysis

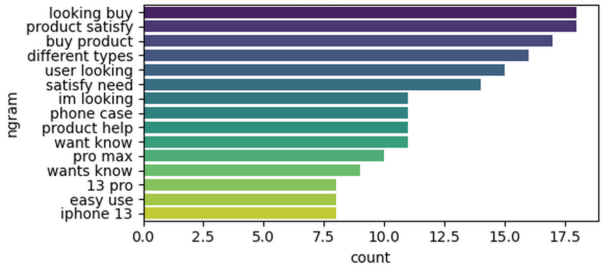
To complement the word-level frequency exploration, we extracted the most frequent **bigrams** (two-word combinations) from each intent category using TF–IDF based tokenization. N-grams provide insight into recurrent lexical patterns, collocations, and phrase structures that characterize the linguistic behavior of each intent. Figure 4.5 presents the top bigrams for the three intent classes.



(a) Top Bigrams *product_details*



(b) Top Bigrams *recommendation*



(c) Top Bigrams *search*

Figure 4.5: Top bigrams per intent class. Frequencies computed on the preprocessed training corpus.

The n-gram analysis highlights recurring lexical dependencies that differentiate the intent categories:

- **Product Details.** Common bigrams such as *faux fur*, *information book*, and *specific product* indicate that utterances in this class often describe tangible attributes or request factual details about a given item. Many phrases contain material descriptors (*fluffy sheepskin*, *ultra soft*), suggesting this intent emphasizes the physical and qualitative aspects of products.
- **Recommendation.** Phrases like *product satisfy*, *buy product*, and *easy use* reveal a preference- or evaluation-oriented language. This confirms that recommendation queries are more subjective, focusing on satisfaction and usability. The presence of overlapping patterns (*looking buy*, *product help*) across both *recommendation* and *search* underscores the semantic proximity between these two intents, which explains the occasional confusion observed during classification.
- **Search.** The most frequent bigrams include *looking buy*, *buy product*, and *different types*. These expressions emphasize exploratory search actions where users attempt to locate or compare items. This pattern reflects typical information-seeking language, characterized by verbs of inquiry (*looking*, *want*, *find*) combined with product-related nouns.

Across all intents, the overlap of terms such as *product*, *buy*, and *use* suggests a shared semantic field centered around purchasing and evaluating products. However, the compositional differences in modifier–head pairs (e.g., *faux fur* vs. *buy product*) clearly separate descriptive from transactional intents. This linguistic evidence supports the later modeling decision to employ transformer-based architectures, which can effectively capture such contextual dependencies beyond the surface-level lexical overlap.

4.2 Multi-label Formulation

While the primary formulation of intent routing in this work is binary, distinguishing between search and recommendation, the CoSRec dataset allows multiple intent labels to be assigned to the same utterance.

As shown in the dataset analysis (3.1), intent categories frequently co-occur, particularly in the case of *search*, which overlaps with both *recommendation* and *product_details*. This observation motivates the exploration of a multi-label formulation, where multiple intents may be predicted simultaneously.

The multi-label setting is not intended to replace the primary task, but rather, we focus on it to provide additional insights into ambiguity and overlapping intent signals.

Setup for Multi-label Classification

In this setting, the task consists of classifying user utterances using the CoSRec dataset, where each utterance may be associated or related with one or more intent labels among *search*, *recommendation*, and *product_details*.

To evaluate robustness and generalization, experiments are conducted across different dataset partitions, including the *Crowd* subset, the high-quality *Curated* subset, and *Synthetic* (pseudo-labeled) data. These configurations allow us to analyze how models behave under varying levels of data quality and diversity.

4.2.1 Results Across Training Data Configurations

Table 4.3: Multi-label classification performance of a logistic regression baseline across different training data configurations.

Training Data	**Class**	**Precision**	**Recall**	**F1**	**Support**
Crowd	Recommendation	0.84	0.84	0.84	219
	Search	0.70	0.83	0.76	227
	Product Details	0.82	0.81	0.81	218
	Macro Avg	0.79	0.83	0.81	664
Crowd + Synthetic	Recommendation	0.99	0.90	0.94	3428
	Search	0.96	0.99	0.97	12357
	Product Details	0.99	0.85	0.92	1835
	Macro Avg	0.98	0.91	0.94	17620
Crowd → Curated	Recommendation	0.81	1.00	0.90	61
	Search	0.40	0.85	0.54	39
	Product Details	0.58	0.94	0.72	36
	Macro Avg	0.60	0.93	0.72	136
Curated	Recommendation	0.79	0.31	0.45	61
	Search	0.33	0.95	0.49	39
	Product Details	0.56	0.42	0.48	36
	Macro Avg	0.56	0.56	0.47	136

As shown in Table 4.3, training on the Crowd subset gives balanced and realistic performance across all intent classes, indicating that diverse human annotations provide robust supervision.

The inclusion of synthetic data significantly increases performance metrics. However, the near-perfect precision and recall suggest that the model is overfitting to synthetic phrasing patterns, resulting so in limited generalization to real conversational data.

To evaluate real-world generalization, models are trained on the Crowd subset and evaluated on the Curated test set. This setting achieves a macro F1 score of 0.72, demonstrating that models trained on diverse, crowd-sourced data generalize effectively to higher-quality annotations, despite a performance drop for more ambiguous intents such as *search*.

In contrast, training and evaluating exclusively on the Curated subset results in substantially lower performance, confirming that the limited size of this dataset is insufficient for

robust model training.

The analysis therefore focuses on the setting where models are trained on the Crowd subset and evaluated on the Curated test set, as this configuration provides the most realistic assessment of generalization.

4.2.2 Feature Analysis for Classical Models

To better understand the role of feature representations, we conduct a systematic evaluation of classical models using different input features.

Table 4.4: Logistic Regression results on Curated test set.

Features	**Class**	**Precision**	**Recall**	**F1-score**
TF-IDF + Bigrams	Rec.	0.83	0.98	0.90
	Search	0.37	0.77	0.50
	Prod.	0.54	0.94	0.69
	Macro	0.58	0.90	0.70
TF-IDF + Context	Rec.	0.78	0.84	0.81
	Search	0.39	0.79	0.53
	Prod.	0.51	0.94	0.66
	Macro	0.56	0.86	0.67
TF-IDF + Handcrafted	Rec.	0.85	0.98	0.91
	Search	0.38	0.77	0.50
	Prod.	0.56	0.94	0.70
	Macro	0.59	0.90	0.70

The results in Table 4.4 show that incorporating bigrams provides modest improvements, particularly for the Recommendation intent, by capturing short multi-word expressions.

Adding conversational context does not yield consistent gains and often reduces precision, suggesting that simple concatenation introduces noise rather than useful signal. In contrast, handcrafted linguistic features lead to the most balanced performance, indicating that coarse semantic cues remain informative for intent classification.

However, across all configurations, performance on the *search* intent remains consistently lower, highlighting the difficulty of modeling this class using sparse lexical representations.

Comparison of Classical Models

To isolate the effect of model choice independently of feature design, we compare alternative classifiers using the same feature representation.

The comparison shows that more complex models such as Random Forests and SVMs do not consistently outperform Logistic Regression. This suggests that, in this setting, performance is primarily limited by feature representation rather than model capacity.

Table 4.5: Performance of classical baseline models under different training data configurations.

Model	**Training Data**	**Class**	**Prec.**	**Rec.**	**F1**	**Supp.**
Random Forest	Crowd → Curated	Rec.	0.87	0.98	0.92	61
		Search	0.36	0.72	0.48	39
		Prod. Det.	0.54	0.92	0.68	36
		Macro Avg	0.59	0.87	0.70	136
SVM	Crowd → Curated	Rec.	0.78	1.00	0.88	61
		Search	0.37	0.74	0.50	39
		Prod. Det.	0.57	0.94	0.71	36
		Macro Avg	0.57	0.90	0.70	136
Logistic Regression	Curated	Rec.	0.93	1.00	0.96	13
		Search	1.00	0.14	0.25	7
		Prod. Det.	0.00	0.00	0.00	8
		Macro Avg	0.64	0.38	0.40	28

Taken together, these results highlight a fundamental limitation of classical approaches: while lexical and handcrafted features capture some intent-specific patterns, they fail to adequately model the ambiguity and variability of *search* queries.

This limitation motivates the transition to contextualized representations. Accordingly, the next stage of this work explores Transformer-based models, including BERT, RoBERTa, and DeBERTa, which are better suited to capturing semantic nuance and overlapping intent signals.

To ensure a fair comparison across models, all Transformer experiments were conducted within a unified training framework. Evaluation metrics (micro- and macro-averaged F1) were computed at each epoch, and the checkpoint achieving the best validation F1 was retained. Further analyses investigate the impact of maximum input length, learning rate selection, and class imbalance handling, with the goal of identifying configurations that yield robust and consistent performance across all intent types.

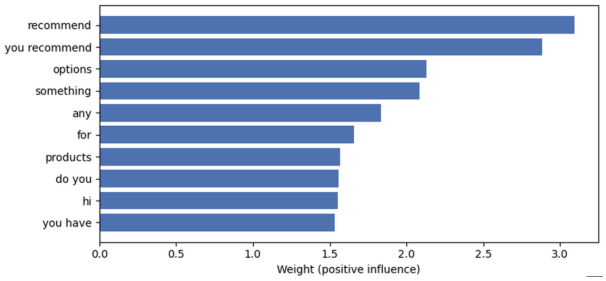


Figure 4.6: Top Features for recommendation

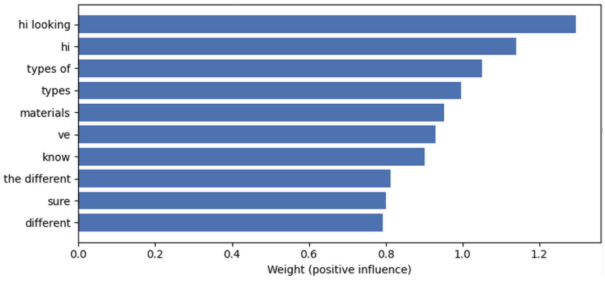


Figure 4.7: Top Features for search

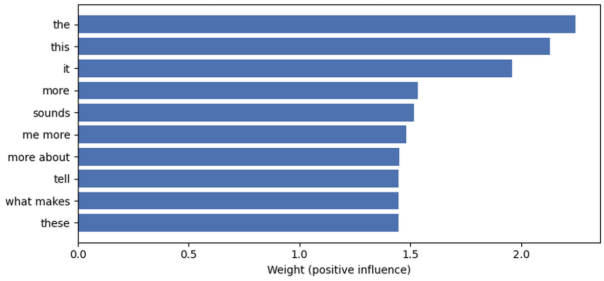


Figure 4.8: Top Features for product_details

Figure 4.9: Top weighted features learned by the Logistic Regression classifier for each intent class. These reflect the lexical patterns associated with each label in the crowd-trained model.

Random Forest Underperformance

Despite being a non-linear ensemble, Random Forest (RF) underperforms both Logistic Regression (LogReg) and Linear SVM. This is mainly due to the text representation (high-dimensional, sparse TF-IDF) and the limited amount of training data per label. Linear models are well suited to sparse data and can leverage global word weights; in contrast, tree-based models must select individual features at each split, which is ineffective when most features are zero and only few positive examples are available. As a result, RF shows lower recall on the rarer label (*product_ details*), which brings down macro-F1.

So the main five key technical reasons for the underperformance are listed below:

- High-Dimensional and Sparse Input Representation: The TF–IDF input is high-dimensional and sparse, a regime where linear models (LogReg, SVM) excel by efficiently learning a single global weight per feature.
- Ineffective Tree Splits on Sparse Features: A decision tree selects a single feature for a split. With tens of thousands of features, a split is often selecting one ”rare word”. The local nature of these splits cannot match the global signal captured by a linear separator.
- Limited Dataset Size for Ensemble Learning: With only a few hundred examples per label, RF cannot learn robust splits.

- Noise and Overfitting on the Rare Multi-Label: For the rarer label (product details), RF encounters few positive nodes, leading to trees that overfit the limited examples, resulting in poor generalization and reduced overall Macro F1.

- Superior Capture of "Topic" Signal by Linear Models: The three intents are highly "topic-like" (e.g., words like "recommend", "specs", "find" are strong class indicators). Linear models excel at assigning large weights to these key tokens, providing a clean, effective linear separation that the tree ensemble cannot replicate

Before the widespread adoption of neural encoders, feature-based text classification methods relied on statistical association measures such as Chi-Squared (χ^2) and Mutual Information to identify discriminative lexical signals in high-dimensional sparse representations.

Comparative studies have shown that these approaches remain effective for intent and topic classification, particularly in settings where utterances are short and token sparsity is high [36, 1].

This line of work provides theoretical grounding for the use of TF-IDF-based representations and motivates the inclusion of linear models as strong and competitive baselines for intent routing.

4.3 Transformer-Based Intent Classification

In this section we will expand the earlier experiments with classical models by introducing Transformer-based methods using the Hugging Face Transformers library, to capture so the deeper nuance for accurate inent routing. Synthetic data was intentionally excluded from the final Transformer runs due to its noise and tendency to overfit to artificial phrasing and to ensure that the findings reflect genuine generalization to human-like interactions.

We fine-tuned pretrained encoders (BERT-base, RoBERTa-base, and DeBERTa-v3-base). Since this is a multi-classification task, a utterance might be assigned to multiple intents all at once so each encoder output was connected to a sigmoid multi-label classification head producing three logits, converting them to probabilities between 0 and 1 one per intent(recommendation, search, and product details). The was optimized using Binary Cross-Entropy with Logits (BCEWithLogitsLoss).

Some intents were underrepresented in the dataset (e.g., Product Details). To compensate for this class imbalance, we applied a class weighting strategy through the *pos_weight* parameter in the Binary Cross-Entropy loss. The *pos_weight* is computed as the ratio of negative to positive samples for each intent, amplifying the contribution of rare classes during training. This ensures that minority intents generate stronger gradient updates and remain visible to the model despite their lower frequency.

Why BCE and not softmax: Softmax assumes mutually exclusive classes whose probabilities sum to 1. Our utterances may express multiple intents simultaneously it would force the model to choose only one, so independent sigmoid activations were necessary.

$$L = -\frac{1}{N}\sum_{i} y_i, \log(\sigma(z_i)) + (1 - y_i), \log(1 - \sigma(z_i)), . \tag{4.1}$$

if yi = 1, the first term penalizes the model if it predicts a small probability, otherwise the second term penalizes the model if it predicts a high probability. The logit part will compute everything(combines sigmoid and BCE) in one go avoiding rounding errors when z is very large or very small, also **Focal BCE** was tested for imbalance, where the loss includes an additional $(1 - p_t)^\gamma$ to reduce the weight of "easy" examples and make the model focus on "hard" or misclassified ones.

4.3.1 Training phase

The transition from classical baselines to Transformer-based architectures requires a specialized training protocol to handle the representational gap between large-scale pre-training and the sparse, telegraphic nature of the CoSRec corpus . The objective of this phase is to develop a robust "Neural Gatekeeper" capable of distinguishing between open-domain information-seeking and personalized recommendation intents . To achieve this, models were fine-tuned using the Hugging Face Trainer API (Text Classification), which was extended to include class-weighted loss functions and fine-grained layer-wise optimization . These extensions are critical for addressing the "linguistic fragility" of the Search intent, ensuring the model does not develop a bias toward the more dominant recommendation cues. To make sure to have stable convergence and prevent catastrophic forgetting during fine-tuning, two complementary strategies were adopted: Layer-wise Learning Rate Decay (LLRD) and partial layer freezing.

Argument	Value	Explanation
max_seq_length	256	Captures enough context for utterances; longer sequences improved recall in ablation (+1–2 F1).
per_device_train_batch_size	8–16	Balance between stability and memory limits.
learning_rate	2e-5	Typical fine-tuning LR for base transformers; stable across backbones.
weight_decay	0.01	Prevents overfitting given small curated validation set.
num_train_epochs	4–5	Converged by epoch 4 for BERT; 5 used for RoBERTa to complete cosine schedule.
warmup_ratio	0.1	Stabilizes the start of fine-tuning before learning rate peaks.
fp16	True	Mixed-precision training to reduce VRAM use.
evaluation_strategy	epoch	Evaluate once per epoch for stability.
save_strategy	epoch	Saves checkpoints per epoch.
load_best_model_at_end	True	Reloads best checkpoint automatically.
metric_for_best_model	eval_loss / eval_macro_f1	Metric for model selection depending on run.
greater_is_better	depends on metric	True for F1, False for loss.

Table 4.6: Common training parameters for all Transformer runs.

Layer-wise Learning Rate Decay (LLRD). Under LLRD, deeper encoder layers are assigned progressively smaller learning rates, typically scaled by a decay factor of 0.9 relative to the layer above. This strategy reflects the hierarchical nature of Transformer representations:

lower layers capture general syntactic and semantic patterns acquired during large-scale pre-training, while upper layers encode task-specific information. By reducing the magnitude of parameter updates in the lower layers, LLRD preserves general linguistic knowledge while allowing the upper layers to adapt more aggressively to intent classification.

Layer Freezing. In addition to LLRD, the bottom 1–2 encoder layers were frozen during the first training epoch and unfrozen thereafter. This temporary freezing prevents large and potentially destructive parameter updates during the early stages of optimization, when gradients can be unstable. Once the model enters a more stable training regime, all layers are unfrozen to allow full end-to-end adaptation.

Table 4.7: Run-specific training configurations for each backbone.

Backbone	**Len**	**Batch**	**Epochs**	**LR**	**LLRD**	**pos_w**	**Focal**	**Freeze N**
BERT-base-uncased	256	16	4	2e-5	0.9	✓	–	1–2
RoBERTa-base	256	8	5	2e-5	0.9	✓	–	1–2
DeBERTa-v3-base	256	8	5	2e-5	0.9	✓	opt.	1–2

To address label imbalance, especially for the Search intent, a positive class weighting *pos_ weight* was applied in the loss function. This up-weights gradients for under-represented or harder-to-learn intents, improving recall without excessively harming precision. Weighted loss proved particularly important for stabilizing training across heterogeneous intent types.

All training runs logged model checkpoints, tokenizer configurations, and optimizer states to ensure full reproducibility.

Evaluation Protocol

As already mentioned in section 4.2, the model performance is evaluated on both validation and test splits of the CoSRec dataset. Final results are reported on the Curated test set, that provides high-quality annotations for reliable evaluation.

Given the multi-label nature of the task, performance is measured using micro-F1, macro-F1, and weighted-F1 scores. Among these, macro-F1 is used as the primary metric, as it reflects balanced performance across intent classes.

To address the limitations of a fixed decision threshold, per-class thresholds are optimized on the validation set by maximizing the F1-score for each intent. This approach improves stability across classes compared to the default threshold of 0.5.

For Transformer-based models, early stopping is based on validation performance, using validation loss for BERT and macro-F1 for RoBERTa and DeBERTa.

Model	micro-F1	macro-F1	weighted-F1
BERT (base)	0.8260	0.8274	0.8279
SVM	0.8124	0.8164	0.8158
Logistic Regression	0.8078	0.8113	0.8102
Random Forest	0.7792	0.7839	0.7841

Table 4.8: BERT vs. classical baselines on curated test set.

DeBERTa for Intent Routing

To address the limitations of standard Transformers, we evaluate DeBERTa (Decoding-enhanced BERT with disentangled attention) [21]. Unlike BERT or RoBERTa, DeBERTa processes content and relative positions as independent vectors. This disentangled mechanism allows the model to better capture subtle semantic relationships, which is crucial for intent routing in conversational systems where queries are often short and underspecified.

Experimental Setup The model is fine-tuned on the CoSRec Crowd subset and evaluated on the Curated test set. We employ Layer-wise Learning Rate Decay (LLRD) to preserve pretrained knowledge while adapting to the multi-label classification task. Per-class threshold optimization is applied based on validation performance to handle label imbalance.

Results and Analysis DeBERTa achieved the highest overall performance, specifically improving macro-F1 over BERT and RoBERTa. The most significant gain was observed in the search intent that is a class traditionally difficult for models relying on lexical overlap.

By modeling contextual dependencies rather than surface-level patterns, DeBERTa effectively disambiguates overlapping classes like search and recommendation. These results confirm that DeBERTa's architectural improvements are better suited for resolving the inherent ambiguity of user intent in joint search and recommendation settings.

4.4 Binary Intent Routing Methodology

In a CSR system, the intent classifier serves as a routing policy. Its primary function is to determine whether a user utterance should be handled by a closed-world recommendation system or by an open-domain search engine (e.g., MS MARCO v2.1). To gain a deeper understanding of how these neural models navigate the boundary between fundamentally different backend actions, we transition the experimental framework to a binary routing formulation (Search vs. Recommendation). This shift is not a mere simplification, but a deliberate design choice motivated by the following system-level requirements. In a joint CSR environment, the Intent Router acts as a gatekeeper for two distinct retrieval worlds: an open-world corpus (MS-MARCO) for general information seeking and a closed-world catalog (Amazon) for personalized suggestions In contrast, the Product Detail intent focuses on attribute inspection for a known item, a task better satisfied through downstream Information Extraction (IE) or Question Answering rather than a high-level routing decision. Isolation of the "Routing Boundary": Early analysis indicates that the most significant semantic ambiguity exists specifically at the interface between Search and Recommendation .

By removing the "Product Detail" intent, which is conceptually orthogonal to the routing mission, we isolate the core modeling challenge: distinguishing between a user's desire for general knowledge and their desire for personalized item suggestions . Ensuring Statistical Uniformity: This methodological pivot creates a uniform (50/50) label distribution across the training and test sets . This balanced environment is essential for a rigorous evaluation of the router's discriminative power, ensuring that performance metrics reflect the model's genuine ability to resolve ambiguity rather than a simple bias toward the most frequent label . By redefining the task as a binary operational gatekeeper, we set the stage for a granular investigation into how different routing policies and contextual signals influence the system's reliability before it ever engages a retrieval engine. This refined setup allows us to systematically analyze the "louder" and "quieter" signals in conversational language without the interference of unrelated task heterogeneity To address this, the original multi-label classification task is reformulated into a binary routing problem under two alternative operational policies.

Dataset Pruning

Before training, all utterances annotated with the *Product Detail* intent were removed (1,988 instances in CoSRec-Crowd). This pruning step is motivated by system-level considerations: Product Detail requests typically require attribute extraction or item-specific fact lookup, which is neither equivalent to open-domain retrieval nor to catalog-based recommendation. Including them would therefore introduce an additional response mode outside the binary routing decision studied in this work.

Label Mapping and Policies

After pruning, labels are mapped into a binary formulation depending on the routing policy.

Model A: Prioritize Recommendation (Recall-Oriented Policy) The Prioritize Recommendation policy is designed to be inclusive and engagement-focused. Under this policy, any utterance containing a hint of recommendation (labeled as {Rec} or {Rec+Search}) is treated as a positive instance. The goal is to maximize user involvement by ensuring that no potential opportunity for personalized suggestion is overlooked.

- Positive class ($y = 1$): {Recommendation}, {Recommendation + Search}
- Negative class ($y = 0$): {Search}

Model B: Prioritize Search (Precision-Oriented Policy) In contrast, the Prioritize Search policy acts as a conservative routing mechanism intended to maximize empirical correctness. Under this formulation, any utterance containing a hint of search is mapped to the positive class, ensuring that informational requests are routed to the document retrieval engine.

- Positive class ($y = 1$): {Search}, {Search + Recommendation}

- Negative class ($y = 0$): {Recommendation}

This mapping forces mixed-intent utterances toward a single operational decision and defines the system's behaviour under ambiguity.

Two input configurations are considered:

- **Context-Only:** user utterance combined with conversational history
- **Context + Profile:** same input augmented with structured user profile information derived from past reviews

Both configurations are trained under identical settings, allowing isolation of the effect of personalization signals on routing decisions.

Tokenization and Input Encoding

The Transformer model uses standard special tokens to encode the input sequence:

- `[CLS]`: classification token (beginning of sequence)
- `[SEP]`: separator token
- `[PAD]`: padding token
- `[UNK]`: unknown token
- `[MASK]`: mask token

User queries and conversational context are concatenated using these tokens to form a single input sequence suitable for Transformer-based classification.

4.4.1 Model Architecture

The intent router is implemented using the `DeBERTa-v2` Transformer architecture via the `DebertaV2ForSequenceClassification` module. Compared to standard BERT-style models, DeBERTa introduces a disentangled attention mechanism that separates content and positional representations, enabling more flexible context modeling. This is particularly relevant for conversational inputs, where subtle changes in phrasing and structure often signal intent differences.

The task is formulated using a multi-label setup with independent sigmoid activations, allowing the model to represent overlapping intent signals rather than enforcing strict mutual exclusivity.

Table 4.9: Configuration of the DeBERTa-based intent router

Parameter Name	Value	Description
model_type	deberta-v2	Transformer architecture
hidden_size	768	Embedding dimension
num_hidden_layers	12	Number of encoder layers
num_attention_heads	12	Attention heads per layer
intermediate_size	3072	Feed-forward layer size
hidden_act	gelu	Activation function
dropout	0.1	Regularization
max_position_embeddings	512	Maximum sequence length
problem_type	multi_label_classification	Independent label prediction

Loss Function

Training employs Binary Cross-Entropy with Logits loss:

$$\ell_i = (1 - y_i)z_i + \log(1 + e^{-z_i})$$

This formulation is appropriate for intent routing, as it allows partial activation of competing intent signals and does not enforce a single dominant class.

To address the relative difficulty of identifying Search intent, a class-weighting strategy is applied using the `pos_weight` parameter, increasing the penalty for misclassifying Search examples.

4.4.2 Optimization and Training Strategy

The fine-tuning of Large Language Models (LLMs) like DeBERTa and BERT for specific downstream tasks such as intent routing requires careful optimization to balance the retention of pre-trained knowledge with the adaptation to task-specific nuances. All models in this study are optimized using the **AdamW** (Adam with Weight Decay) optimizer [**?**], which decouples weight decay from the gradient update, leading to better generalization compared to standard Adam.

The primary hyperparameter configuration is as follows:

- **Learning rate:** 2×10^{-5}, selected to provide a stable descent on the loss landscape without overshooting local minima.
- **Batch size:** 16, balancing computational efficiency with the stochasticity needed for robust gradient estimation.
- **Number of epochs:** 5, a duration found to be sufficient for convergence based on the parameter sensitivity analysis in Section 5.3.

- **Warmup ratio:** 0.1, which linearly increases the learning rate during the initial 10% of training steps to prevent gradient instability in the early stages of fine-tuning.

To further improve training stability and mitigate the risk of **catastrophic forgetting**, where the model loses its general linguistic capabilities while over-optimizing for the narrow CSR dataset, two advanced techniques are applied:

- **Layer-wise Learning Rate Decay (LLRD):** We apply a decay factor of 0.9 across the transformer layers. This ensures that lower layers (which capture fundamental syntax and semantics) undergo smaller updates, while higher layers (which capture task-specific intent) are allowed to adapt more significantly.
- **Gradual Unfreezing (Layer Freezing):** During the first epoch, the lower layers of the encoder are frozen, focusing the backpropagation exclusively on the classification head and the top-most transformer blocks. These layers are subsequently unfrozen for the remaining epochs. This "staged" approach allows the newly initialized classification parameters to stabilize before the pre-trained weights are modified.

These strategies ensure that the model preserves the robust representations learned during pre-training while effectively navigating the high-dimensional objective of the Joint Search and Recommendation task.

Mixed-Intent Evaluation Setup

To explicitly study ambiguity, a subset of 26 utterances originally annotated with both Search and Recommendation intents is isolated. These samples are excluded from training and used exclusively for evaluation, enabling controlled analysis of model behaviour under mixed-intent conditions.

Summary

This methodology defines a controlled framework for studying intent routing under competing operational policies. By combining dataset reformulation, structured input representations, and Transformer-based classification, the approach enables systematic investigation of routing behaviour, ambiguity handling, and the interaction between personalization and intent prediction.

Chapter 5

Experiments Results and Analysis

This chapter presents the experimental evaluation of intent routing models on the CoSRec dataset. We begin by establishing baseline performance under the multi-label formulation, comparing classical machine learning models and Transformer-based architectures. We then analyze model behavior at a finer level, focusing on per-label performance, error patterns, and probabilistic characteristics. The goal is not only to compare models, but to understand the sources of ambiguity that affect routing decisions in conversational settings.

5.1 Overall Results

We first evaluate overall classification performance to establish a high-level comparison across model families. Table 5.1 reports micro, macro, and weighted F1 scores on the CoSRec Curated test set, where we can observe that transformer-based models consistently outperform classical baselines, confirming so the importance of contextual representations for intent classification in short conversational utterances. Among them, DeBERTa achieves the best performance across all metrics with an F1-score of 83.6%, followed by BERT with 82.8% and the RoBERTa with 82.1%.

Classical models such as SVM and Logistic Regression remain competitive, particularly in terms of macro-F1, due to their effectiveness in high-dimensional sparse TF-IDF feature spaces. However, they exhibit a consistent performance gap compared to Transformer models. In contrast, Random Forest performs substantially worse, suggesting that tree-based methods struggle to generalize in sparse and high-variability text settings.

Table 5.1: Overall micro/macro/weighted F1 on the CoSRec Curated test set.

Model	**Micro F1**	**Macro F1**	**Weighted F1**
DeBERTa	0.8317	0.8352	0.8356
BERT	0.8260	0.8274	0.8279
RoBERTa	0.8191	0.8194	0.8207
SVM	0.8124	0.8164	0.8158
LogReg	0.8078	0.8113	0.8102
RF	0.7792	0.7839	0.7841

While these aggregate results establish a clear performance hierarchy, they do not fully capture routing behavior under ambiguity. In particular, overall F1 scores, because aggregate F1 scores act as a "global comparison" that obscures significant variations across intent labels and hides the specific failure modes of the system. For this reason, we proceed with a more detailed per-label and probabilistic analysis.

because aggregate F1 scores act as a "global comparison" that obscures significant variations across intent labels and hides the specific failure modes of the system.

5.1.1 Per-label Performance and Error Analysis

This section evaluates the model's performance on a label-by-label basis. By analyzing individual results, we identify the specific intents that lead to misclassifications. This granular view is helps for understanding how the model handles ambiguous or rare user requests. Table 5.2 reports per-label F1 scores for these three Transformer-based models.

The *recommendation* intent is consistently the easiest to classify, with F1 scores approaching 89% across all architectures. This suggests that recommendation-oriented utterances contain relatively stable and discriminative lexical cues, such as evaluative language and explicit references to products or use cases.

In contrast, the *search* intent exhibits the lowest performance, with F1 scores remaining around 77%. Errors for this class are dominated by false negatives, indicating that search utterances are frequently misrouted to other intents. This pattern is consistent across all models, suggesting that the difficulty is not model-specific but instead reflects intrinsic properties of the intent itself.

The *product_details* intent occupies an intermediate position. Although it benefits from the contextual modeling capabilities of Transformers , most notably DeBERTa it remains sensitive to class imbalance and semantic overlap with recommendation oriented requests. Attribute-heavy utterances often trigger confusion between these two intents.

Table 5.2: Per-label F1 on the test set (Transformers).

Label	**BERT**	**RoBERTa**	**DeBERTa**
recommendation	0.8821	0.8927	0.8918
search	0.7742	0.7692	0.7749
product_details	0.8259	0.7962	0.8390

The gap on *search* indicates that queries phrased as navigational or exploratory requests are short, varied, and sometimes ambiguous; models over-predict *product_ details* when the utterance contains explicit attributes (e.g., sizes, colors), and over-predict *recommendation* for comparative phrasing. DeBERTa's relative gains on *product_ details* suggest improved handling of attribute-heavy phrases.

5.1.2 Probabilistic Metrics (AUC/AP)

Accuracy and F1-score alone do not capture the quality of confidence estimates, which are critical for intent routing decisions. We therefore evaluate ranking-based and probabilistic metrics that are threshold-independent. [18, 14]

To assess ranking quality independent of decision thresholds, we compute micro/macro ROC–AUC and Average Precision (AP). Table 5.3 reports threshold-independent ranking metrics. DeBERTa achieves the highest macro and micro ROC–AUC as well as Average Precision, indicating superior separability and more stable ranking behaviour across intent classes.

Table 5.3: Threshold-independent ranking metrics across transformer backbones. Higher values indicate better separability and ranking quality.

Model	**ROC–AUC (macro)**	**ROC–AUC (micro)**	**AP (macro)**	**AP (micro)**
BERT	0.86	0.88	0.89	0.91
RoBERTa	0.87	0.89	0.90	0.92
DeBERTa	**0.89**	**0.90**	**0.91**	**0.93**

The reduction in ROC-AUC when profile information is added suggests increased overlap between class score distributions, indicating that user profiles inject a recommendation-oriented bias. [18] [6]

Confusions and Error Patterns

We inspect confusion matrices for each label under the tuned thresholds. Typical error modes are:

- **Recommendation ↔ Product Details.** Utterances asking for similar items but mentioning specific attributes (e.g., "like this one in black") can trigger both of the intents. The model may favor *product_ details* due to the presence of attributes.

- **Search misses.** Very short or telegraphic search queries (e.g., single nouns) tend to be under-detected, hurting recall in *search*.

Table 5.4: Sample misclassified utterances (DeBERTa). The model's mistakes largely reflect semantic overlap between intents and very short queries.

Text (truncated)	**True labels**	**Predicted labels**
"show me something similar in black size M"	{recommendation, product_details}	{product_details}
"nike air" (telegraphic query)	{search}	{recommendation}
"what are the specs for this model"	{product_details}	{recommendation}

Calibration and ranking-based metrics such as ROC–AUC are particularly useful for diagnosing failure modes in intent routing systems, where confidence misalignment can lead to catastrophic policy decisions [14, 6].

All three transformers performed much better than the traditional ones, classical baselines. This proves that the models are better at understanding the hidden meaning behind words, even when the user types a very short message. Among these, DeBERTa was the top performer because it is excellent at separating the "meat of a sentence, like specific product features, from the "glue" words, like "the" or "and".

The biggest problem remains the Search intent; the system often misses these requests because they are short and easily drowned out by more dominant language. To fix this, we need to provide the model with more diverse examples to learn from or train it specifically to recognize when someone is trying to look up information . Based on the final experiments, using these augmentation techniques successfully fixed this bottleneck and made the system much more reliable

5.2 The Personalization Paradox in Intent Routing

While the initial results in Section 5.1 provide a baseline for general intent classification, a functional CSR system requires a specific routing decision: should the system trigger a Search engine or a Recommender? To analyze this critical 'junction' point, we refine the task into a binary routing problem. Personalization is a cornerstone of conversational recommendation systems, where user history and long-term preferences are essential for ranking suitable items. In contrast, upstream intent routing within a Joint Conversational Search and Recommendation (CSR) system serves a different function: it must first determine whether a personalized recommendation is required at all, or whether the user request should be routed to an open-domain search engine.

5.2.1 Personalization Bias and Systematic Failure

Prior work has shown that personalization can introduce unintended bias when dominant personal signals override the immediate intent expressed in a query. Teevan et al. report that while personalization may benefit ambiguous queries, it can degrade performance when user-specific information "swamps" the more general intent shared across users [48].Empirical analysis on the CoSRec dataset reveals an extreme manifestation of this effect. User profiles in CoSRec are represented as textual summaries derived from past reviews. When these profiles are injected into a DeBERTa-based intent router under a Search-priority policy, the model exhibits catastrophic failure. Specifically, classification accuracy drops from 0.894 (Context-Only) to 0.116 (Context + Profile), while ROC-AUC collapses to 0.061, indicating a nearly complete loss of discriminative ability. The Matthews Correlation Coefficient [31] becomes strongly negative, demonstrating systematic misclassification rather than random error.This failure underlines a critical architectural awareness: while personalization is crucial for downstream item ranking, it is damaging to upstream intent routing. Profile information introduces a convincing recommendation-oriented prior that swamps the routing decision, even when the current utterance asserts an informational need. These findings motivate an exact separation between routing and ranking components, where intent classification depends solely on conversational context, and personalization is applied only after the routing decision has been made.

5.2.2 Parameter Sensitivity and Learning Dynamics

To determine whether the observed performance limitations are due to suboptimal training or inherent properties of the data, we conduct a systematic hyperparameter sensitivity analysis. Specifically, we vary the learning rate, batch size, and number of training epochs.

Each parameter provides insight into a different aspect of the learning process. If performance improves with more epochs, this would indicate underfitting and insufficient training. If performance is sensitive to learning rate, this would suggest optimization instability or difficulty in navigating the loss landscape. Similarly, strong variation across batch sizes would indicate sensitivity to gradient noise and training dynamics.

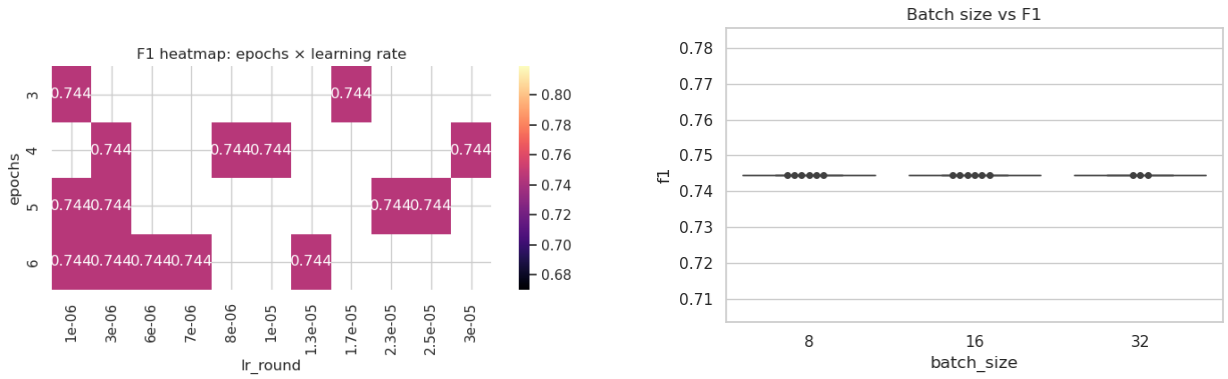
Conversely, if performance remains stable across all configurations, this would suggest that the model has already reached its optimal capacity under the given input representation, and that further optimization is unlikely to yield improvements.

The goal of this analysis is therefore to distinguish between optimization limitations and intrinsic limitations of the data.

Mainly this was done to understand whether optimization choices meaningfully affect the model's ability to separate Search and Recommendation intents or if the "ceiling" is intrinsic to the data itself. Across all tested configurations, the model exhibits remarkably stable behavior. As seen in Figure 5.1, the F1-score consistently fluctuates within a narrow range of approximately 0.74–0.75 and does not exceed this bound under any hyperparameter combination.

The following observations characterize the learning dynamics:

- **Rapid Convergence:** Increasing the number of epochs from three to six produced no



(a) F1 heatmap across learning rates and epochs. (b) F1 score as a function of different batch sizes.

Figure 5.1: Hyperparameter sensitivity analysis. Performance remains stable across learning rates and training epochs, indicating limited sensitivity to optimization parameters.

statistically meaningful improvement. This suggests that the model reaches its optimal decision boundary early in training and that underfitting is not a dominant issue.

- **Gradient Stability:** The F1-score remains effectively constant across batch sizes of 8, 16, and 32, suggesting that batch-induced noise does not significantly impact the learning process for this task.
- **Fine-tuning Robustness:** Comparable performance is observed across learning rates ranging from 10^{-6} to 3×10^{-5}, indicating that the process is robust to learning rate selection and that catastrophic forgetting is unlikely within this regime.

Synthesis: The Data Signal Ceiling

Taken together, these findings indicate that the binary intent classifier is not optimization-limited. Instead, it encounters a clear performance ceiling around F10.74, which persists regardless of hyperparameter tuning. This ceiling, coupled with the "swamping" effect of profile data described in Section 5.2.1, strongly suggests that the limiting factor is the *intrinsic separability* of the intents within the dataset.This observation is critical for the remainder of this analysis. It confirms that the bottleneck is not the training procedure, but the input signal itself. Consequently, subsequent sections explore alternative strategies, such as data augmentation and ambiguity-focused analysis, to improve performance by enhancing the clarity of the conversational context rather than further tuning optimization parameters.

5.3 Ambiguity Analysis: Lexical and Probabilistic Asymmetry

Mixed-intent utterances represent the most challenging cases in a Conversational Search and Recommendation (CSR) system, as they simultaneously express both information-seeking

and recommendation-oriented signals. Rather than treating these cases as noise, we analyze them as a controlled setting to probe the model's internal biases and understand how routing decisions are made under ambiguity.

From the pruned binary dataset, we isolate a subset of 26 utterances originally annotated with both Search and Recommendation intents. These queries are excluded from training and used exclusively for analysis. Two DeBERTa-based routers are evaluated: Model A (Recommendation-priority) and Model B (Search-priority). For each model, we analyze the predicted probability assigned to its respective priority class.

5.3.1 Probabilistic Behaviour on Mixed-Intent Utterances

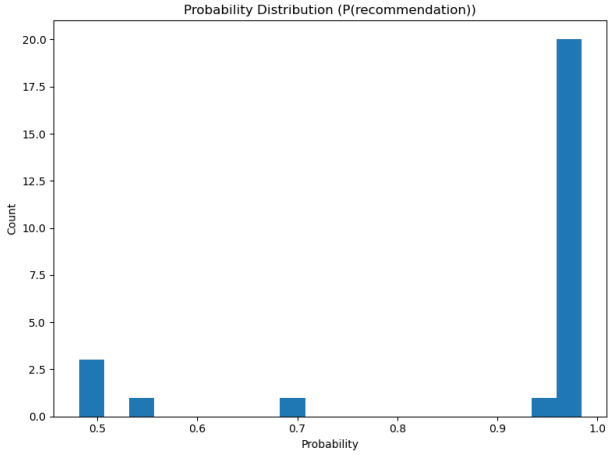


Figure 5.2: Predicted probability distributions on the mixed-intent subset. Model A outputs $P(\text{Rec})$, while Model B outputs $P(\text{Search})$.

Figure 5.2 reveals a clear contrast in probabilistic behaviour between the two policies. Under the Recommendation-priority policy, the distribution is sharply bimodal, with most utterances assigned high confidence scores close to $P(\text{Rec}) \approx 1.0$. Only a small fraction lies near the decision boundary.

In contrast, the Search-priority model produces a much flatter distribution, with predictions spread across a wide range and many clustered around the decision threshold. This indicates substantially higher uncertainty when the model is required to prioritize Search.

This asymmetry is reflected in the summary statistics. Model A achieves $\overline{P}(\text{Rec}) = 0.895$ with low variance ($\sigma = 0.174$), while Model B achieves $\overline{P}(\text{Search}) = 0.614$ with higher dispersion ($\sigma = 0.272$).

At the standard threshold $\tau = 0.5$, Model A classifies 92.3% (24/26) of mixed-intent utterances as Recommendation, whereas Model B assigns only 53.8% (14/26) to Search, a behaviour close to random assignment. This suggests that mixed-intent queries lie near the decision boundary under a Search-priority formulation.

Geometric Evidence of Semantic Dominance

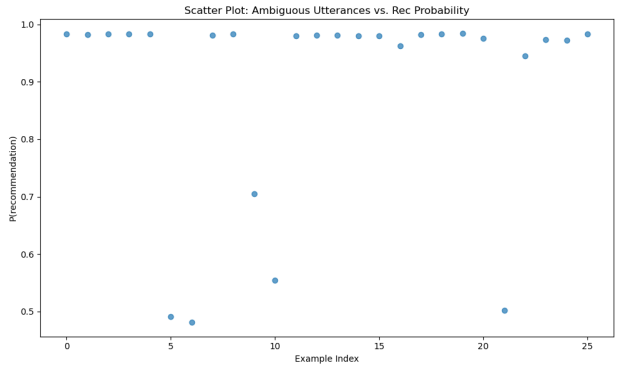


Figure 5.3: Scatter plot of predicted probabilities for mixed-intent utterances.

Figure 5.3 provides a geometric interpretation of this asymmetry. The dense cluster of points near $P(\text{Rec}) \approx 0.95$–$1.0$ demonstrates that ambiguous utterances lie substantially closer to the Recommendation region in the learned representation space.

This indicates that, even when both intents are present, the surface form of the query aligns more strongly with recommendation semantics—typically characterized by product categories, attributes, and implicit comparisons— rather than with the simpler and more open-ended structure of Search queries.

5.3.2 Lexical Drivers of Asymmetry

To explain this probabilistic behaviour, we analyze the lexical patterns associated with high- and low-confidence predictions. The mixed-intent utterances are partitioned into high-confidence ($p > 0.9$) and low-confidence ($p < 0.6$) groups.

Tokens Driving High Recommendation Confidence

Table 5.5: Key tokens associated with high Recommendation confidence.

Token Category	**Examples**	**Interpretation**
Product attributes	*rubber, premium, heavy-duty*	Strong descriptive features for item selection
Product categories / brands	*car, sneakers, jeep, cherokee*	Concrete anchors in product space
User constraints	*walking, arch, support*	Preference-driven evaluative signals

These tokens define a constrained solution space and strongly resemble catalog-oriented requests. As a result, the model confidently maps such utterances to the Recommendation class.

Tokens Driving Low Confidence (Search Signals)

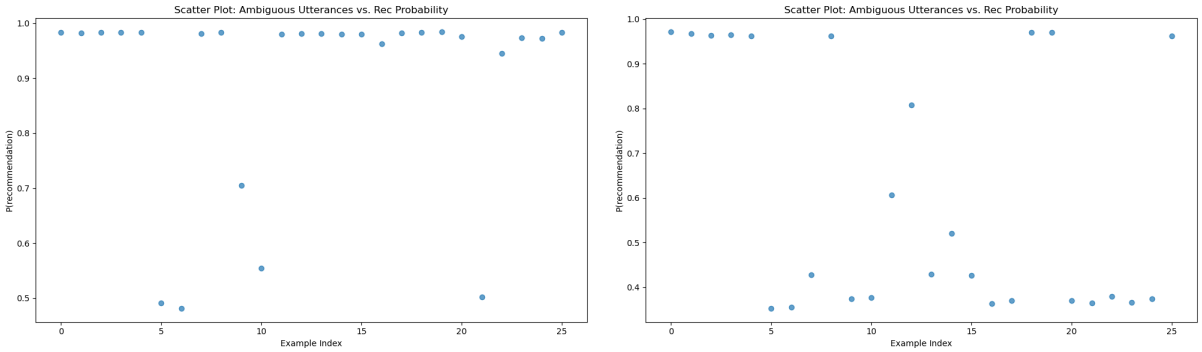
Table 5.6: Key tokens associated with ambiguous or Search-oriented signals.

Token Category	Examples	Interpretation
Interrogatives	*do, which, have*	Explicit information-seeking intent
Comparative terms	*difference, between*	Open-ended comparison
Technical nouns	*headlight, materials*	Specification-oriented queries

These cues are weaker and less constraining, allowing multiple possible interpretations. This pulls predictions toward the decision boundary, increasing uncertainty.

Summary of linguistic asymmetry. Recommendation intent is supported by "loud" lexical signals that define a closed candidate space, while Search intent is expressed through shorter, more variable, and structurally weaker language. This asymmetry explains the observed probabilistic imbalance.

Cross-Model Comparison



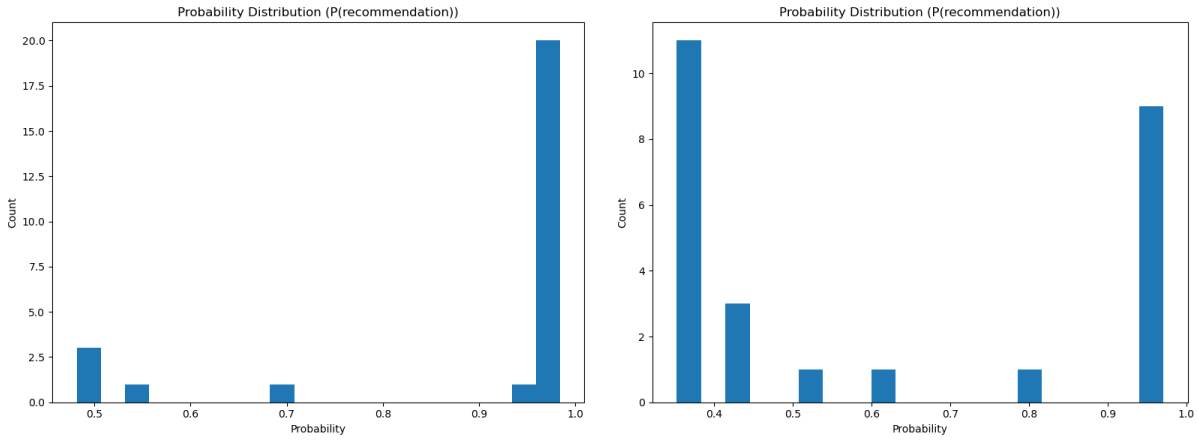
(a) Recommendation-priority (b) Search-priority

Figure 5.4: Scatter comparison across routing policies.

The asymmetry is in the Figures 5.4 is shown as evident. Recommendation-priority produces dense clustering at high confidence of ($p > 0.9$), whereas Search-priority shows a wide spread across $[0.35, 0.95]$, confirming that ambiguous queries are closer to the Recommendation manifold then Search.

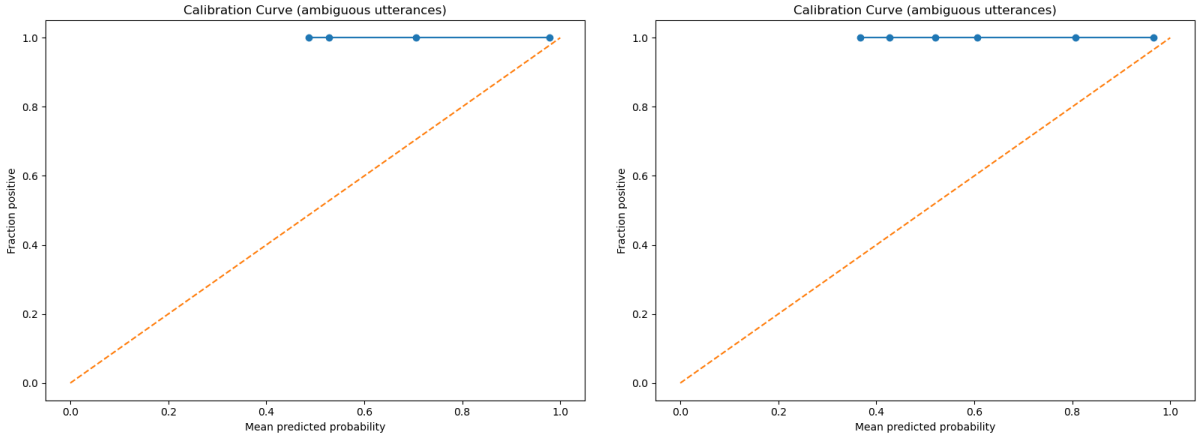
The histograms reinforce this: Recommendation predictions form a sharp peak, while Search predictions remain diffuse and uncertain.

Calibration analysis shows that Recommendation-priority predictions align closely with true frequencies, while Search-priority predictions are poorly calibrated in the mid-range, indicating unstable confidence.



(a) Recommendation (b) Search

Figure 5.5: Probability distributions across policies.



(a) Recommendation-priority (b) Search-priority

Figure 5.6: Calibration comparison across policies.

This analysis leads to three key conclusions:

1. **Ambiguity is semantic, not noise.** Mixed-intent queries systematically combine two linguistic patterns rather than reflecting annotation error.

2. **Recommendation dominates.** Most ambiguous queries lie closer to Recommendation in the latent space.

3. **Search is the fragile class.** It is lexically weaker, more variable, and associated with higher uncertainty.

These findings justify treating routing as a policy decision rather than a purely accuracy-driven classification task. Recommendation-priority aligns with the natural dominance of recommendation language, while Search-priority provides a conservative alternative for minimizing misrouting of informational queries.

Utterance	P(Rec)	P(Search)	Cues
"premium rubber floor mats for jeep cherokee"	0.97	0.41	Attributes, brand (Rec)
"difference between depo and tyc headlight assembly"	0.48	0.71	Comparison (Search)
"comfortable sneakers with arch support"	0.96	0.39	Preferences (Rec)
"do car wallets have keychains"	0.52	0.64	Interrogative (Search)

Table 5.7: Representative ambiguous utterances.

Overall, this ambiguity analysis provides a principled explanation for model behaviour, linking probabilistic outputs to underlying linguistic structure.

5.4 Strategic Routing Policies

In this section we analyse the routing policies already introduced in Section 4.4. The two operational strategies are a *Recommendation-priority* (recall-oriented) policy and a *Search-priority* (precision-oriented) policy.

5.4.1 Recommendation-Priority Policy (Model A)

The Recommendation-priority policy is designed to maximize user engagement by treating any utterance containing recommendation signals as a positive instance. This includes both pure Recommendation and mixed-intent queries.

As for quantitative performance, Table 5.8 reports the performance under two input settings. Incorporating user profile information improves overall metrics, including Accuracy, Precision, and MCC, while slightly reducing Recall.

Table 5.8: Recommendation-Priority Model Performance Comparison

Metric	**Context-Only**	**Context + Profile**
Accuracy	0.7611	0.8123
Precision	0.7469	0.8178
Recall	0.9581	0.9162
F1-score	0.8394	0.8642
MCC	0.4508	0.5731
ROC-AUC	0.9412	0.9150

Table 5.8 reports the core classification metrics for the Recommendation-Priority policy under the two input settings: **Context-Only** and **Context + Profile**. Under this policy, the positive class corresponds to RECOMMENDATION, and mixed-intent utterances are also mapped to Recommendation. This means that the model is intentionally encouraged to capture any utterance that contains even partial recommendation-like evidence.

The first important observation is that the **Context-Only** model achieves extremely high recall (0.9581). This is a central result, because recall is the most policy-aligned metric in a recommendation-first routing system. In practical terms, recall measures how many true recommendation-oriented utterances are successfully captured by the router. A recall of 0.9581 means that the model misses very few recommendation opportunities. This is highly desirable if the system objective is to maximize engagement and avoid losing users who may be seeking personalized suggestions.

However, this strong recall comes with a trade-off. The **Precision** of the Context-Only model is lower (0.7469), meaning that although the model captures almost all recommendation cases, it also incorrectly routes a substantial number of search-oriented utterances into the recommendation class. This creates a typical recall–precision trade-off: the model is highly inclusive, but this inclusiveness causes over-prediction of Recommendation.

When user profile information is added, the system becomes more balanced. The **Context + Profile** model improves in Accuracy (0.8123 vs. 0.7611), Precision (0.8178 vs. 0.7469), F1-score (0.8642 vs. 0.8394), and MCC (0.5731 vs. 0.4508). These improvements indicate that profile information helps the model become more selective and reduces unnecessary recommendation assignments. In other words, the profile acts as a moderating signal that reduces the model's tendency to interpret every ambiguous product-related utterance as recommendation intent.

At the same time, Recall decreases slightly from 0.9581 to 0.9162. This reduction is small but meaningful. It suggests that profile information makes the classifier more conservative. For a recommendation-priority system, this may or may not be desirable depending on the system objective. If the thesis research prioritizes *not missing recommendation opportunities*, then the Context-Only model is more aligned with the routing goal. If instead the aim is to achieve a more balanced classifier with fewer false recommendation predictions, then the Context + Profile setting becomes more attractive.

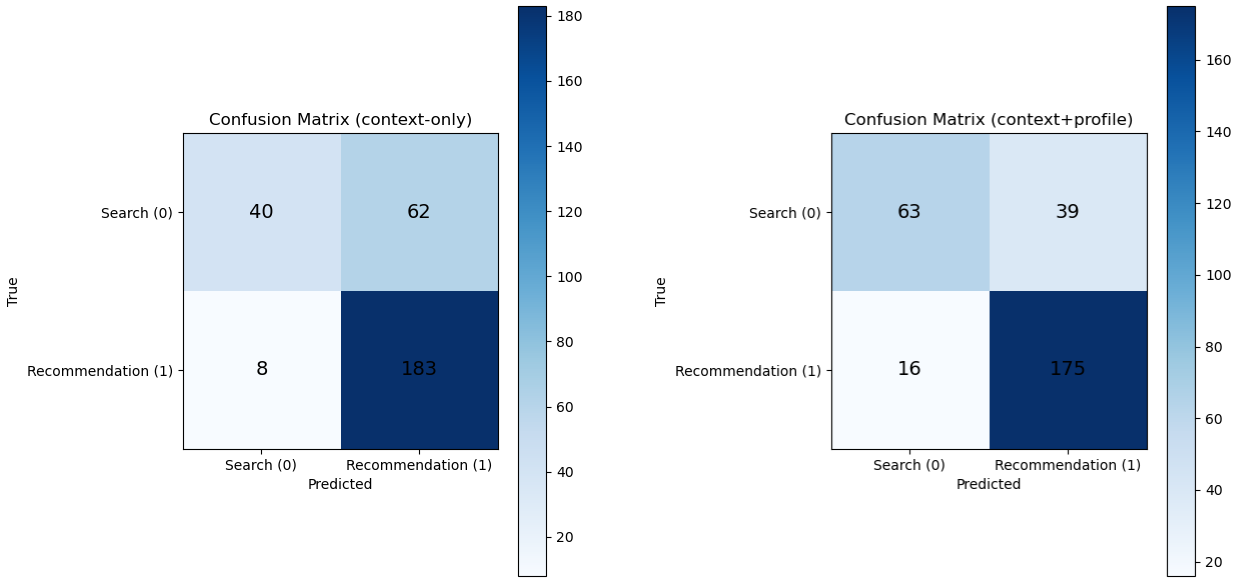
The F1-score and MCC help explain this balance further. Since F1-score combines precision and recall, and MCC summarizes classification quality while accounting for all four

confusion matrix cells, the fact that both improve under Context + Profile shows that the profile feature improves overall operational stability. Thus, the quantitative results establish an important methodological insight for the thesis: user profiles can improve aggregate classification quality in a recommendation-first system, but they do so by sacrificing some of the extreme recall bias that defines the policy itself.

This directly affects the thesis research goal because the thesis is concerned with how to build intent routers that align with distinct system-level objectives. The Recommendation-Priority model demonstrates that metrics must be interpreted relative to policy intent. A higher F1-score is not automatically "better" if it reduces the router's ability to capture recommendation opportunities. Therefore, the research shows that model evaluation must be policy-aware rather than purely metric-maximizing.

Error structure and confidence behaviour. The Context-Only configuration aggressively defaults to Recommendation, resulting in a high number of false positives (62 Search utterances misclassified). Adding user profiles reduces these errors (down to 39), indicating improved decision boundaries.

Probability distributions reveal strong overconfidence in Recommendation predictions, particularly under the Context-Only setting. Profile information partially mitigates this effect by redistributing predictions toward mid-range probabilities and improving calibration.



(a) Context-Only (b) Context + Profile

Figure 5.7: Confusion matrices for the Recommendation-Priority model.

Figure 5.7 presents the confusion matrices for the Recommendation-Priority model under both input settings. These matrices are critical because they reveal the *structure of the model's errors*, not just their quantity.

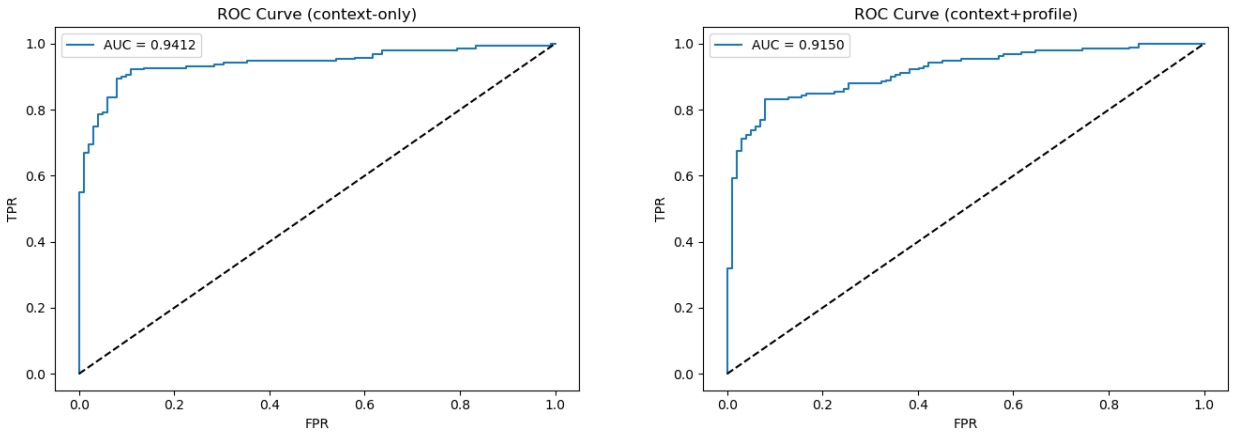
For the **Context-Only** model, the confusion matrix shows a strong bias toward Recommendation. The most notable result is that **62 Search utterances are misclassified as**

Recommendation. This confirms that the model aggressively defaults to the positive class whenever an utterance contains lexical or semantic cues associated with products, attributes, or evaluative language. This behavior is fully consistent with the routing policy: the model is designed to be inclusive and to absorb ambiguous cases into Recommendation. From a policy perspective, this is not necessarily a flaw. Instead, it is evidence that the model has learned the intended operational bias.

Nevertheless, the confusion matrix also reveals the cost of this design. Search utterances that include product-specific vocabulary may be incorrectly treated as recommendation requests, even when the user is actually seeking factual or comparative information. This means that the router may sometimes send informational queries to the recommendation backend, which could reduce answer relevance and harm user satisfaction in those cases.

The **Context + Profile** confusion matrix shows a clear improvement in this respect. The number of Search utterances incorrectly routed as Recommendation falls from **62 to 39**. This is one of the most important empirical effects of the profile signal in the Recommendation-Priority setting. It indicates that profile information helps the model "hesitate" before classifying an utterance as Recommendation. In practical terms, the user profile adds contextual information that constrains overgeneralization.

This matters for the thesis because it demonstrates that confusion matrices reveal policy behavior more clearly than scalar metrics alone. The thesis research goal is not merely to produce a classifier with a high score, but to understand how routing policies behave under ambiguity. The confusion matrix analysis shows that Recommendation-first routing naturally produces asymmetric errors, especially false positives on Search. This supports the broader thesis claim that recommendation language is semantically dominant and tends to absorb nearby intent classes.



(a) Context-Only (AUC = 0.9412) (b) Context + Profile (AUC = 0.9150)

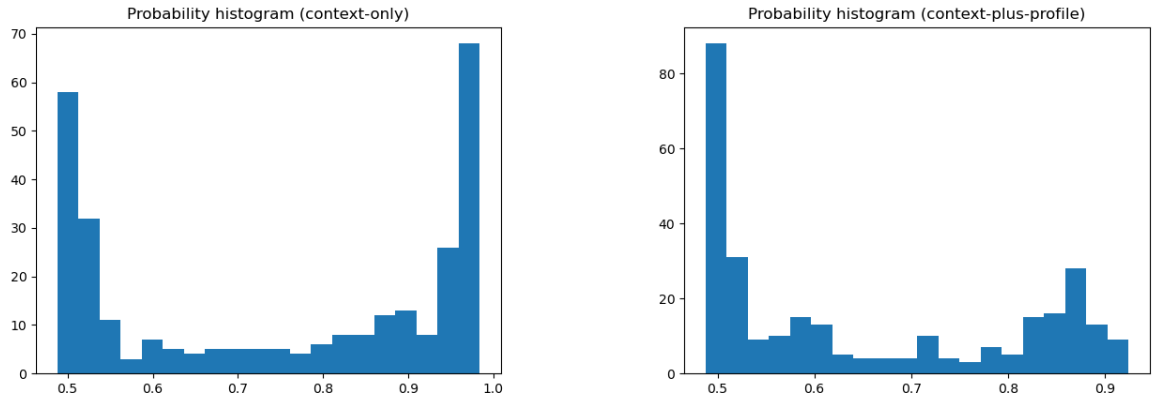
Figure 5.8: ROC comparison for Recommendation-Priority model.

Figure 5.8 compares the ROC curves for the two Recommendation-Priority models. The **Context-Only** variant achieves a ROC-AUC of 0.9412, while the **Context + Profile** variant achieves 0.9150.

At first glance, this may seem surprising, because the profile-enhanced model performs better on several threshold-based metrics such as Accuracy, Precision, F1-score, and MCC. However, ROC-AUC measures something different: it evaluates the model's ability to *rank* positive cases above negative cases across all possible thresholds. The fact that the Context-Only model has the higher AUC suggests that, at a representation level, its class separation is actually stronger.

This implies that the inclusion of profile information introduces additional overlap between Recommendation and Search probability distributions. In other words, the profile appears to help with final decisions at the chosen operating threshold, but it slightly weakens the model's global ranking separability. This is an important distinction. It shows that the profile is not simply making the model "better" in a universal sense. Instead, it changes the geometry of the decision space: it smooths and moderates the classifier, improving some practical metrics while slightly weakening global separability.

For the thesis, this is methodologically significant. It demonstrates that performance cannot be understood through a single metric. A model may show lower ROC-AUC yet better thresholded decision-making, depending on how auxiliary inputs shift the probability distribution. This supports the thesis argument that routing should be evaluated through a combination of discrimination, calibration, and policy-consistent error analysis.



(a) Context-Only (b) Context + Profile

Figure 5.9: Probability histograms for Recommendation-Priority predictions.

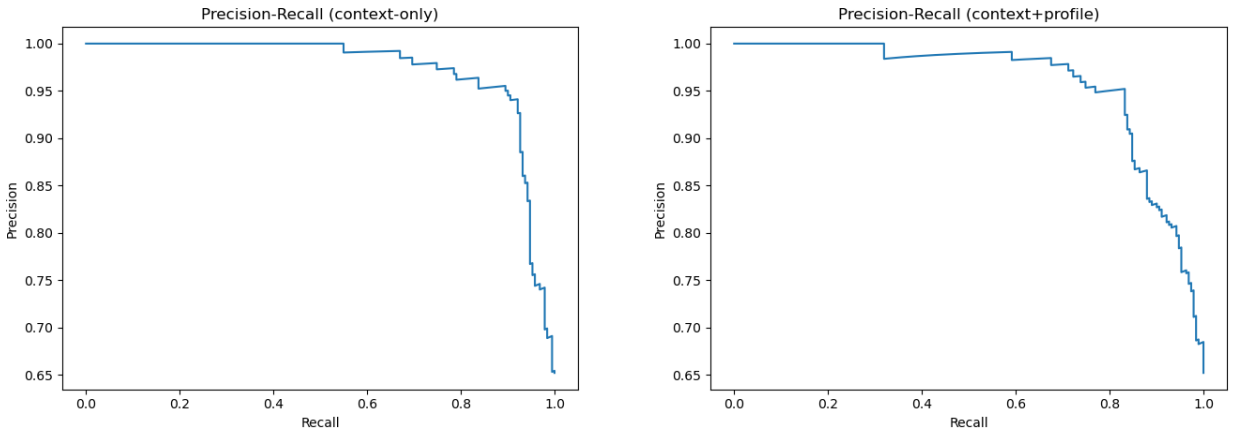
Figure 5.9 shows the distribution of predicted Recommendation probabilities. In both input settings, the distribution is strongly bimodal, meaning that most examples are assigned either very high or very low Recommendation probability rather than values near the decision threshold.

This is an important finding because bimodality generally indicates that the classifier sees many utterances as clearly belonging to one class or the other. For the **Context-Only** model, however, the histogram is heavily skewed toward $P(\text{Rec}) \approx 1.0$. This means that once the model detects product-related or evaluative language, it often becomes extremely confident that the utterance belongs to Recommendation. Such over-concentration near 1.0 reflects what the thesis identifies as the **semantic dominance of recommendation**

language. Words such as "best," "comfortable," "premium," and references to products or personal use cases are highly salient cues that activate the recommendation manifold.

When the profile is included, some of these extremely confident predictions shift toward more moderate probability ranges. This indicates that the profile reduces overconfidence and helps the model distribute belief more cautiously. From a methodological perspective, this is valuable because it shows that user profile information does not simply alter correctness, but also changes *confidence behavior*. This matters in routing systems, because overly confident models are less reliable under ambiguity and harder to calibrate for downstream thresholding.

For the thesis research goal, the histogram analysis strengthens the claim that Recommendation is the more lexically and semantically stable intent class. Recommendation utterances produce loud signals, while Search signals are weaker and easier to overshadow. This asymmetry is central to the thesis and is directly visible in the shape of the prediction distributions.



(a) Context-Only (b) Context + Profile

Figure 5.10: Precision–Recall curves for Recommendation-Priority model.

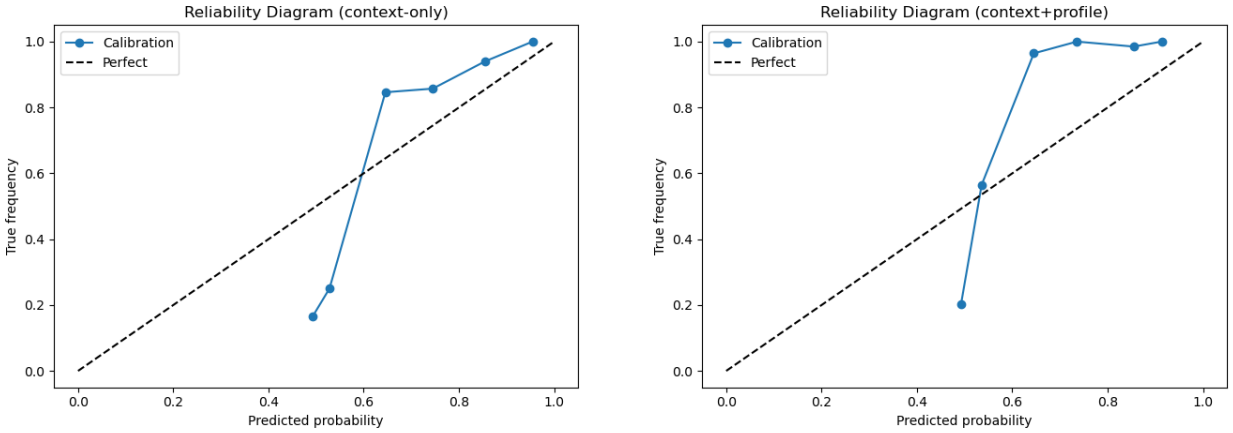
Figure 5.10 presents the Precision–Recall curves for the Recommendation-Priority model. These curves are especially informative because they show how precision changes as recall increases across decision thresholds.

A key result reported in the text is that **precision remains nearly perfect (at or above 95%) across most recall values**. This tells us that Recommendation intent is highly identifiable when explicit recommendation-style language is present. In other words, when the model predicts Recommendation under strong lexical evidence, it is usually correct.

This is highly relevant to the thesis because it demonstrates that Recommendation is not only the dominant class under the policy mapping, but also the more semantically coherent one. Its boundary is easier to learn and more robust to threshold variation. This contrasts with Search, which later appears much more fragile in threshold-based evaluation.

From the perspective of the research goal, the PR curve supports the conclusion that recommendation-first routing is naturally well suited to conversational systems where engagement and suggestion coverage are primary goals. It also implies that recommendation-like

phrasing provides a rich and stable signal that can be exploited effectively by a transformer-based router.



(a) Context-Only (b) Context + Profile

Figure 5.11: Reliability diagrams for Recommendation-Priority model.

Figure 5.11 shows the calibration behavior of the Recommendation-Priority model. Calibration refers to how closely predicted probabilities correspond to empirical outcome frequencies. For example, if the model assigns many utterances a probability of 0.9, calibration asks whether roughly 90% of those utterances are truly positive.

The Recommendation-Priority model is described as **consistently overconfident**, especially in the Context-Only setting. This means that the model often assigns very high Recommendation probabilities more often than is justified by actual correctness. Such overconfidence is a predictable consequence of semantic dominance: when the model sees strong product or evaluative cues, it commits strongly to Recommendation.

When profile information is added, the calibration improves. The reliability curve moves closer to the diagonal, meaning that predicted probabilities better reflect true frequencies. This is important because calibration affects how trustworthy the model's confidence estimates are, especially for downstream decision-making or fallback strategies. A well-calibrated model is more useful in real systems because its probabilities can be interpreted more meaningfully.

In the context of the thesis, this matters because the research is not only about raw classification accuracy, but about designing reliable intent routers for conversational recommendation systems. Calibration analysis reveals whether the model "knows what it knows." The finding that Context + Profile improves calibration while Context-Only maximizes recall shows another policy-relevant trade-off: the most aggressive recommendation-first model is not the most probabilistically reliable one.

Mixed-intent behavior. As shown in Table 5.10, the model assigns high probability to Recommendation in the majority of ambiguous cases.

Table 5.9: Model A (Recommendation-Priority) on Mixed-Intent Utterances

Statistic	**Value**	**Interpretation**
$P(\text{Rec}) > 0.5$	92.3%	Strong default to Recommendation
Mean $P(\text{Rec})$	0.895	High confidence
Std Dev	0.174	Stable predictions

Table 5.10 analyzes how Model A behaves on explicitly mixed-intent utterances. These examples are especially important because they are the most policy-sensitive and theoretically ambiguous part of the dataset.

The model predicts Recommendation with probability above 50% for 92.3% of mixed-intent cases (24/26 utterances). The mean predicted Recommendation probability is 0.895, with a relatively low standard deviation of 0.174. Even the minimum score, 0.481, lies very close to the decision boundary. Together, these statistics show that mixed-intent utterances are overwhelmingly absorbed into Recommendation space.

This result directly supports one of the central claims of the thesis: ambiguous conversational requests are not symmetric between Search and Recommendation. Instead, mixed-intent utterances more strongly resemble Recommendation, both lexically and semantically. Thus, when the policy maps mixed utterances to Recommendation, the model aligns naturally with the data distribution and behaves confidently.

This is highly important for the thesis research goal because it explains why recommendation-first routing is comparatively stable. The model is not forcing the data into an unnatural boundary; rather, it is following the dominant semantic structure of the utterances themselves. This provides a theoretical justification for why recommendation-priority policies are easier to train and more robust in practice.

Table 5.10: Model A (Prioritize-Rec) Performance on Mixed-Intent Utterances

Statistic	**Value**	**Interpretation**
$P(\text{Rec}) > 0.5$	92.3% (24/26)	Strong default to Recommendation
Mean $P(\text{Rec})$	0.895	High confidence
Std Dev	0.174	Stable predictions
Min $P(\text{Rec})$	0.481	Near decision boundary

This behavior reflects the semantic dominance of recommendation language, where evaluative and product-centric tokens (e.g., “best”, “premium”, “comfortable”) act as strong signals.

5.4.2 Search-Priority Policy (Model B)

The Search-priority policy adopts a conservative routing strategy, prioritizing Search whenever an utterance contains any informational signal. This policy aims to minimize incorrect personalization by ensuring that exploratory queries are routed to the retrieval system.

For quantitative performance, Table 5.11 summarizes the main classification metrics for the Search-Priority model and shows that the Context-Only configuration achieves the best balance, with high precision (0.971), strong F1 (0.914), and MCC close to 0.8.

The **Context-Only** model achieves Accuracy = 0.894, Precision = 0.971, Recall = 0.864, F1-score = 0.914, MCC = 0.787, and ROC-AUC = 0.950. These are strong results and show that a transformer-based router can indeed learn a robust Search-first policy when trained on conversational context alone.

Table 5.11: Search-Priority Model Performance

Metric	**Context-Only**	**Context + Profile**
Accuracy	0.894	0.857
Precision	0.971	0.969
Recall	0.864	0.806
F1-score	0.914	0.880
MCC	0.787	0.724
ROC-AUC	0.950	0.955

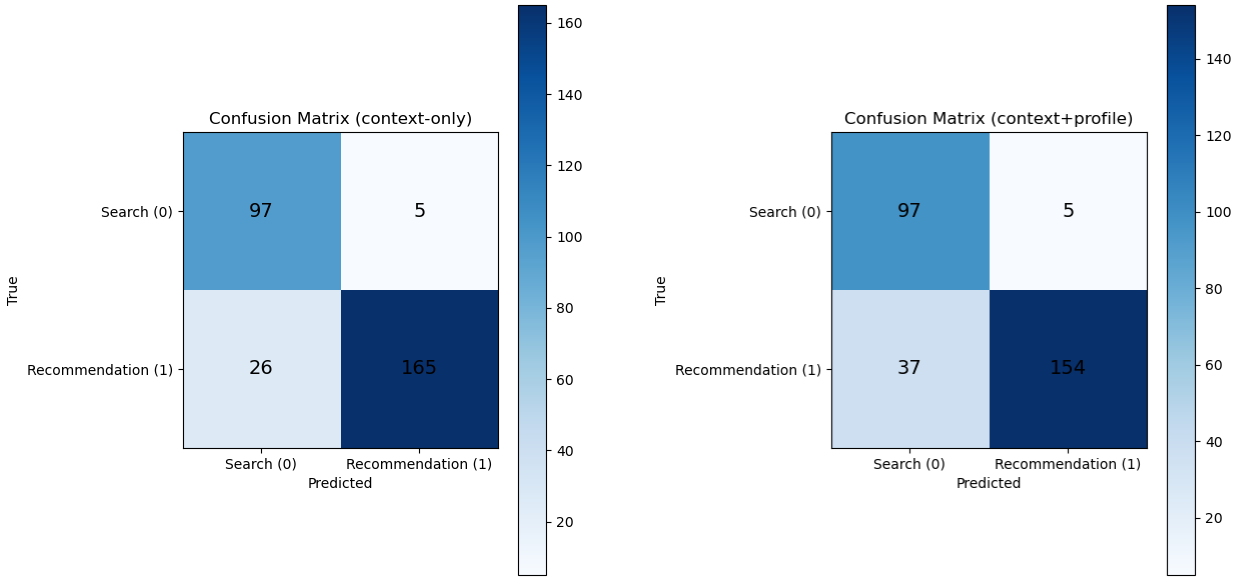
The most important number here is the very high **Precision** of approximately 0.97. Because Search is the positive class, this means that when the model predicts Search, it is almost always correct. This is exactly what a Search-priority policy is designed to achieve. In a real routing system, such precision is critical because it prevents factual or exploratory requests from being lost in the recommendation pipeline.

The **Recall** of 0.864 is also strong, although lower than precision. This means that some true Search utterances are still missed and routed elsewhere, but the majority are successfully identified. The combination of high precision and high recall produces an F1-score of 0.914, which indicates excellent overall balance under the Context-Only setting.

When user profile information is added, Accuracy declines to 0.857, Recall drops to 0.806, F1-score falls to 0.880, and MCC decreases to 0.724, while Precision remains very high at 0.969. This pattern is extremely revealing. It shows that profile information does *not* cause the model to become generally confused or random. Instead, it introduces a **directional bias**: the model remains precise when it predicts Search, but it becomes less willing to predict Search at all. As a result, more true Search utterances are missed.

This has deep implications for the thesis. It shows that personalization cues are architecturally misaligned with Search-first routing. User profiles contain long-term preference information that resembles Recommendation semantics, so when they are injected upstream, they pull the decision boundary away from Search. Therefore, even though profile information may appear harmless in aggregate, it systematically undermines the policy objective of safely capturing informational intent.

Despite similar ROC-AUC values, adding user profiles degrades recall and MCC, indicating that ranking ability does not translate into reliable routing.



(a) Context-Only (b) Context + Profile

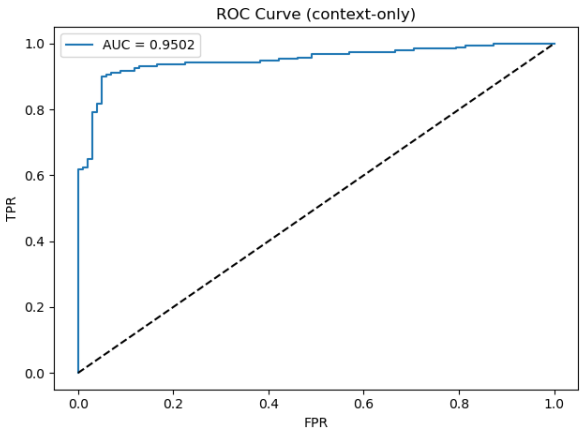
Figure 5.12: Confusion matrices for the Search-Priority model.

Figure 5.12 presents the confusion matrices for the Search-Priority model. These figures clarify how the policy behaves in practice.

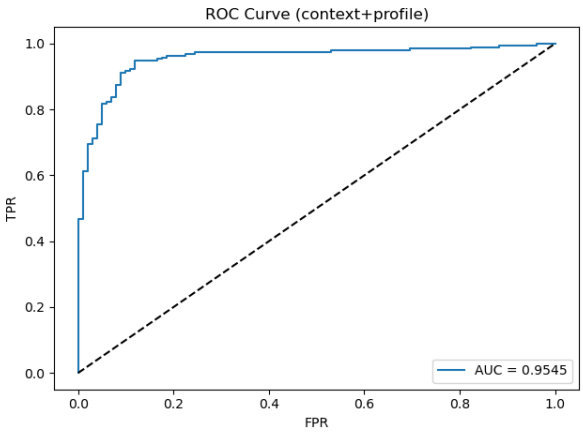
In the **Context-Only** setting, the confusion matrix shows that the model correctly recognizes most Search utterances and rarely misclassifies Recommendation utterances as Search. This is the ideal error profile for a precision-oriented router. It means that when the system decides to send a request to the retrieval engine, that decision is usually justified. The remaining mistakes are mainly *false negatives*, where true Search utterances are not captured. This kind of error is expected because many Search requests are short, underspecified, and semantically close to recommendation language.

When the **Profile** is added, false negatives increase. More genuine Search queries are redirected toward Recommendation. This finding is one of the strongest pieces of evidence in the thesis against using profile information in upstream routing. The confusion matrix does not show a general collapse in the sense of random confusion; rather, it shows a systematic semantic pull toward the Recommendation side.

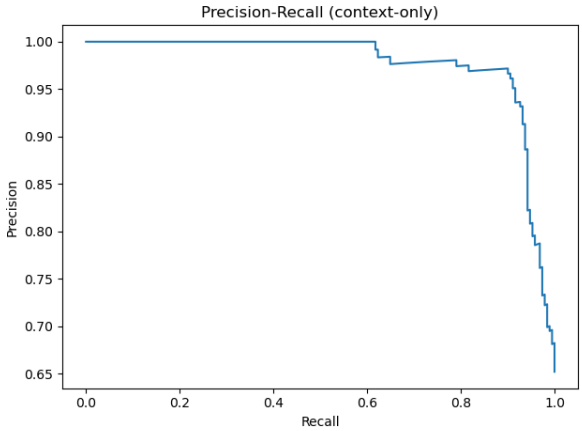
For the thesis research goal, this is highly significant. The objective is to understand how routing policies behave under ambiguous conversational inputs. The confusion matrix proves that Search intent is not simply "harder" in a general sense, but specifically vulnerable to being overshadowed by recommendation-like context. This validates the thesis argument that Search is the semantically fragile class in open conversational routing.



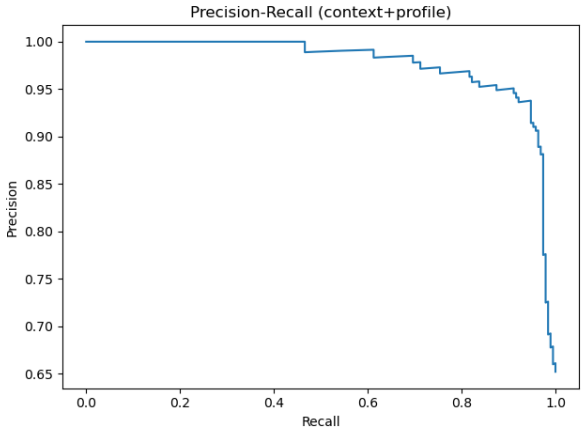
(a) ROC, context-only



(b) ROC, context+profile



(c) PR, context-only



(d) PR, context+profile

Figure 5.13: ROC and Precision–Recall curves for Search-Priority model.

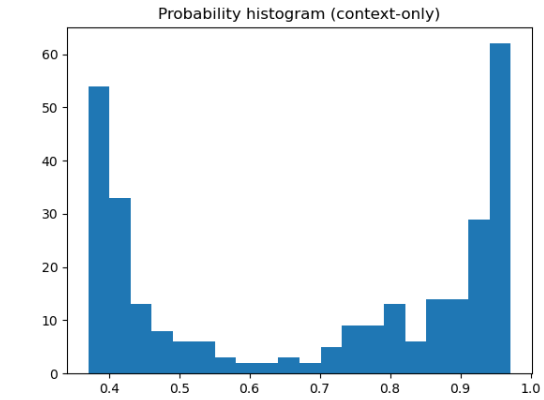
Figure 5.13 reports ROC and PR curves for the Search-Priority model. Both Context-Only and Context + Profile models achieve ROC-AUC values near 0.95, which indicates strong separability between Search and non-Search utterances at a ranking level.

This is important because it shows that the model learns a useful latent distinction between the two classes even in the more difficult Search-first formulation. In other words, the transformer is capable of representing Search intent successfully, despite its lexical sparsity and ambiguity.

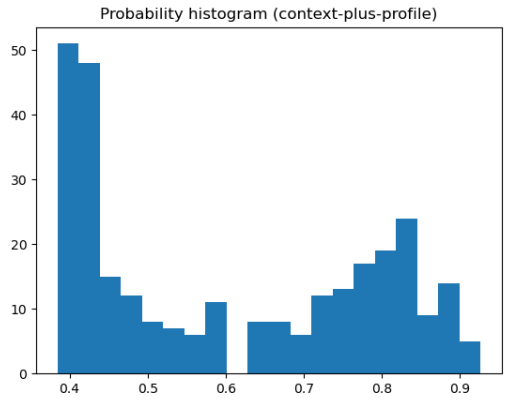
However, the Precision–Recall curves show a more nuanced picture. The **Context-Only** model maintains precision above 0.95 across a large recall range, up to approximately 0.8. This matches the strong quantitative results and confirms that the model can identify Search reliably across many thresholds.

The **Context + Profile** PR curve is slightly flatter, with precision declining earlier as recall increases. This indicates that profile information harms the operational robustness of Search classification. Even though the model still ranks examples reasonably well overall, its threshold-sensitive behavior becomes less stable.

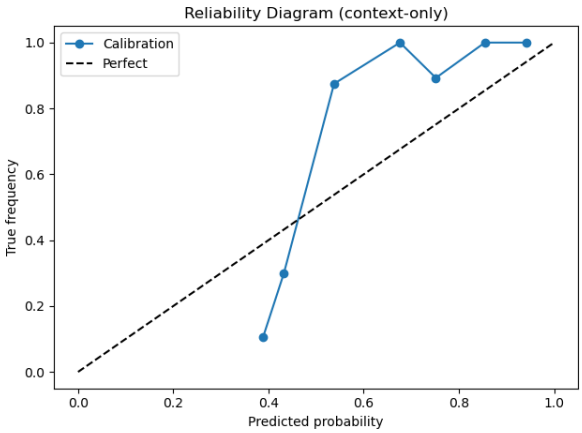
For the thesis, this distinction is crucial. ROC-AUC alone might suggest that the profile-enhanced model is almost as good as the Context-Only model, or even slightly better in ranking terms. But the PR curve reveals that the practical routing behavior has degraded. This supports a central thesis argument: **global separability does not guarantee policy-consistent operational performance.** A routing model must be evaluated where it actually operates, not only across abstract ranking thresholds.



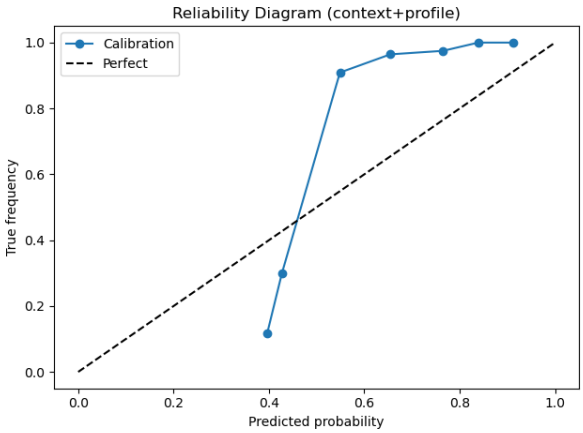
(a) Histogram, context-only



(b) Histogram, context+profile



(c) Reliability, context-only



(d) Reliability, context+profile

Figure 5.14: Probability distributions and calibration for Search-Priority model.

Figure 5.14 reports both probability histograms and reliability diagrams for the Search-Priority model. The histograms show a pronounced bimodal structure in both settings, meaning that many utterances are assigned either low or high Search probability. This is generally a desirable pattern because it suggests that the model is often decisive rather than uncertain.

However, the more important result comes from the calibration analysis. In the **Context-Only** setting, predicted probabilities align reasonably well with empirical frequencies. This

indicates that the model's confidence estimates are relatively trustworthy. High-confidence Search predictions correspond to truly informational utterances with good consistency.

In contrast, when profile information is added, calibration worsens in the middle probability range, particularly around $p \approx 0.4$–0.6. The model becomes **under-confident** in this region: it assigns moderate Search probabilities to utterances that are in fact more strongly Search-oriented. This explains why recall falls. Many utterances that should cross the Search threshold fail to do so because the probability scale has been distorted by recommendation-oriented profile signals.

This is an important conceptual finding for the thesis. It means that the profile does not merely shift labels; it distorts the *meaning* of confidence itself. In a routing system, this is dangerous because confidence is often used for thresholding, fallback rules, and uncertainty handling. A miscalibrated Search router may appear cautious, but actually be systematically biased away from Search.

Thus, the calibration analysis reinforces one of the core claims of the thesis: user profiles should be excluded from upstream Search-priority routing because they are not neutral context features; they are semantic priors that reshape the probability landscape in a policy-incompatible way.

Instability under ambiguity. Unlike Model A, the Search-priority model exhibits significant uncertainty when handling mixed-intent queries. As shown in Table 5.12, predictions cluster near the decision boundary.

Table 5.12: Model B (Search-Priority) on Mixed-Intent Utterances

Statistic	**Value**	**Interpretation**
$P(\text{Search}) > 0.5$	53.8%	Near-random preference
Mean $P(\text{Search})$	0.614	Low confidence
Std Dev	0.272	High uncertainty

Table 5.12 examines the Search-Priority model on mixed-intent utterances. These cases are the most difficult and most theoretically informative, because the policy explicitly forces them toward Search.

The results show that the model predicts Search with probability above 0.5 in only **53.8%** of cases (14/26 utterances). The mean Search probability is 0.614, which is much lower than the corresponding mean for Model A in the Recommendation-first setting. The standard deviation is 0.272, indicating a large spread in confidence, and the minimum probability falls to 0.352, clearly below the decision threshold.

These numbers show that the model is not confidently absorbing mixed utterances into Search space. Instead, it oscillates near the decision boundary and often fails to commit. This is one of the most important empirical results in the thesis because it reveals the **asymmetry of mixed-intent routing**. Mixed utterances are not equally compatible with both policies. They naturally resemble Recommendation more than Search, so forcing them into Search creates uncertainty and instability.

This directly affects the thesis research goal. The thesis aims to understand whether routing policy can change model behavior in principled ways. The mixed-intent analysis shows that policy is not merely a labeling convention. It interacts with the semantic structure of language. Recommendation-first routing aligns with the natural form of mixed utterances, whereas Search-first routing must fight against the dominant lexical signal. This explains why Search-priority routing requires stronger intervention, such as augmentation and careful training stabilization.

This reflects a fundamental *semantic asymmetry*: search queries are short, underspecified, and lexically weaker than recommendation queries. Exploratory tokens (e.g., "difference", "how", "between") are often overshadowed by product-related language.

Calibration and error patterns. The Context-Only model exhibits well-calibrated probabilities and a clear bi-modal distribution, indicating confident separation between classes. In contrast, adding profile information introduces miscalibration, particularly in mid-probability ranges, leading to increased false negatives for Search.

Impact of User Profiles

Across both policies, user profiles act as a strong recommendation-oriented prior. While they slightly improve balance in Recommendation-priority models, they are detrimental under Search-priority settings.

In extreme cases, profile injection leads to catastrophic failure, with ROC-AUC collapsing to 0.061 and MCC becoming strongly negative. This indicates systematic misclassification rather than random error.

These findings highlight a critical architectural constraint: personalization signals are fundamentally incompatible with upstream intent routing. Instead, they should be applied exclusively in downstream ranking stages.

This analysis demonstrates a clear asymmetry between Recommendation and Search intents in conversational routing. Recommendation intent is semantically dominant, producing strong lexical signals and highly confident predictions, which makes the Recommendation-Priority policy naturally stable and recall-efficient. In contrast, Search intent is inherently more fragile, characterized by shorter, less explicit language and greater uncertainty, making the Search-Priority policy more sensitive to ambiguity and external bias.

A key finding is the detrimental effect of user profile information on upstream routing, particularly under the Search-Priority setting. While profiles can improve balance and calibration in recommendation-oriented models, they introduce a strong recommendation bias that distorts decision boundaries and reduces the model's ability to correctly identify informational queries.

Overall, the results confirm that intent routing should rely exclusively on conversational context, while personalization signals should be reserved for downstream recommendation stages. This separation ensures both robust intent detection and effective personalization, aligning the system design with the underlying semantic structure of user queries.

5.5 Reformulating Intent Routing: From Multi-label to Binary

This section focuses on a binary intent classification setting (*Recommendation* vs. *Search*), which forms the basis for a robust and interpretable routing mechanism within a conversational search and recommendation (CSR) system. While the original multi-label formulation includes a third intent, *Product Details*, empirical and system-level analysis motivates its exclusion from the routing task.

From a functional perspective, Product Details requests do not correspond to either of the two backend actions considered in this work: open-domain document retrieval (Search) or closed-catalog item recommendation. Instead, they require attribute inspection or factual lookup of a known product, representing a distinct interaction mode that is orthogonal to routing decisions.

Early multi-label experiments further indicate that the principal modeling challenge lies in distinguishing *Search* from *Recommendation*, with Search consistently achieving lower F1 scores. Including Product Details in this setting obscures this core difficulty and inflates aggregate performance metrics without improving routing reliability. For these reasons, the problem is intentionally reduced to a binary formulation, enabling a focused analysis of routing behavior between Search and Recommendation backends.

In this refined binary phase, the methodology emphasizes three strategic aspects:

- Rebalancing positive and negative examples to ensure fair learning,
- Selecting an operating threshold via precision–recall analysis, and
- Reporting probabilistic metrics, including ROC–AUC and Average Precision, alongside standard precision and recall.

5.5.1 Dataset Distribution

A preliminary inspection of the dataset splits confirms that the binary intent classification task is not affected by class imbalance. Across the training, validation, and test partitions, the distribution between the two target intents remains close to uniform, with approximately 50% of the utterances labeled as *Recommendation* and 50% as *Search*. The partitioning strategy follows a consistent 80%/10%/10% split, ensuring statistical comparability between the subsets.

This balanced distribution is a desirable property for intent routing tasks, as it prevents artificial inflation of performance metrics that can arise when one intent dominates the dataset. In imbalanced scenarios, classifiers may achieve high accuracy by favoring the majority class, masking poor performance on the minority intent. By contrast, the balanced nature of the CoSRec-derived binary dataset ensures that precision, recall, and F1-score reflect genuine model discrimination capabilities rather than label frequency effects.

From a system design perspective, this balance is particularly important. The intent router operates upstream of two fundamentally different backend modules: an open-domain retrieval engine and a closed-catalog recommendation engine. Misrouting either intent carries

a tangible user-facing cost. A balanced dataset therefore allows the evaluation to faithfully reflect real-world trade-offs between factual correctness and recommendation engagement, rather than biasing the system toward one operational mode.

Finally, the consistency of label proportions across all splits supports the reliability of comparative experiments conducted in later sections. Performance differences observed across models, hyperparameters, or routing policies can be attributed to modeling and representation choices rather than sampling artifacts.

5.5.2 Analysis of Intent Classification

The strategic classification and analysis phase explicitly leverages mixed-intent utterances rather than discarding them. These samples represent realistic user behavior in CSR systems, where informational and recommendation needs frequently co-occur within a single turn.

Two complementary experimental strategies are employed:

- **Prioritized Binary Classification.** Ambiguous utterances are incorporated into training under two alternative routing policies: a Recommendation-priority policy, where any hint of recommendation is treated as positive, and a Search-priority policy, where any hint of search is treated as positive. This design allows the analysis of which intent signal is more robustly captured by the model under competing system objectives.

- **Deep Dive into Ambiguity.** A core binary classifier trained exclusively on pure Search and pure Recommendation utterances is evaluated on the mixed-intent subset. This setting isolates the intrinsic linguistic overlap between the two intents and enables a fine-grained analysis of uncertainty, confusion boundaries, and probabilistic behavior.

Together, these strategies establish a controlled and domain-specific framework for analyzing intent routing behavior. They provide insight into how contextual Transformer models, such as DeBERTa, distinguish between open-domain information-seeking and personalized recommendation intents within a unified CSR architecture.

While the original CoSRec framework is designed for multi-intent labeling, the goal in this thesis is to analyze the capability of Transformer-based language models to *route* user requests to the correct backend subsystem: either the product recommendation engine (closed catalogue) or the open-domain document retrieval module (MS-MARCO v2.1).

To support this, a cleaned binary dataset was constructed following the CoSRec-Crowd distribution. All utterances containing the *Product Detail* intent, which requires a fundamentally different system response (product-level information extraction) were removed. The remaining 2,280 Search + Recommendation labeled utterances were used to build two different priority-based datasets:

- **Prioritize Recommendation:** utterances with {Rec} or {Rec+Search} receive label 1; pure Search receives label 0.

- **Prioritize Search:** utterances with {Search} or {Search+Rec} receive label 1; pure Recommendation receives label 0.

This reflects the real-world system routing policies where ambiguous user requests must be forced toward one backend based on design preference.

The analysis presented in this chapter focuses on the **Recommendation-priority dataset**, using DeBERTa v3 Base as the underlying encoder. Two evaluation settings were then compared:

1. **Context-Only**, where the classifier receives the user utterance concatenated with conversational history.
2. **Context + Profile**, where the model additionally receives the user profile text extracted from CoSRec-Curated.

The purpose of this comparison is to determine whether user profile information enhances the classifier's ability to distinguish between Search and Recommendation intents, or whether it introduces bias due to the structure of the CoSRec profiles.

5.5.3 Mixed Intents in CSR

The previous experiments showed that DeBERTa-v3-based classifiers reach high performance on the binary *Search vs. Recommendation* task, but also revealed a stable performance ceiling and a systematic asymmetry between the two classes. To understand whether the remaining errors are due to noisy labels or to genuinely overlapping semantics, we perform an *ambiguity deep dive* focused on the utterances that were annotated with both Search and Recommendation intents in CoSRec.

Experimental Setup

We start from the pruned binary dataset described in Section 5.5, where all *Product Detail* intents are removed and only Search / Recommendation occurrences are retained. From this pool we identify the subset of utterances whose intent annotations contain *both* types simultaneously, for example requests that both ask for general information and express a preference for item suggestions. After mapping these to utterance-level query variants, we obtained a set of 26 *mixed-intent* queries.

For a deeper dive, neither model is retrained, instead, both are applied *only* to the 26 mixed-intent queries. For each query we record the predicted probability $p = P_\theta(y{=}1)$ for the "priority" class (Recommendation for Model A, Search for Model B). We then inspected how many ambiguous utterances are classified as the priority class under different thresholds $\tau \in \{0.5, 0.6, 0.7\}$, to then compute descriptive statistics over the predicted probabilities, and finally perform a lexical analysis of the tokens that dominate high-confidence versus low-confidence predictions.

Mixed-Intent Prediction Behaviour

Table 5.13 reports how the two models behave when forced to make a binary decision on the 26 mixed-intent queries.

Model	Threshold	#Rec	#Search	%Rec	%Search
Prioritize-Rec	$\tau = 0.5$	24	2	92.3%	7.7%
	$\tau = 0.6$	22	4	84.6%	15.4%
	$\tau = 0.7$	22	4	84.6%	15.4%
Prioritize-Search	$\tau = 0.5$	12	14	46.2%	53.8%
	$\tau = 0.6$	11	15	42.3%	57.7%
	$\tau = 0.7$	10	16	38.5%	61.5%

Table 5.13: Predictions on the 26 mixed-intent utterances (containing both Search and Recommendation labels) under different decision thresholds τ. "#Rec" denotes the number of utterances classified as the priority class (Recommendation for Model A, Search for Model B).

The Prioritize-Rec model classifies almost all mixed-intent utterances as Recommendation: at the default threshold $\tau = 0.5$, 24 out of 26 queries (92.3%) are routed to the Recommendation class, and this remains above 84% even when the threshold is increased to 0.7. The probability distribution is highly skewed towards Recommendation, with a mean predicted probability $\overline{p}_{\text{rec}} = 0.895$, standard deviation 0.17, and a minimum of 0.48. Entropy-based uncertainty is correspondingly low (mean entropy ≈ 0.21), and only 15.4% of the mixed utterances fall into a low-confidence region ($p < 0.6$); 80.8% are classified with very high confidence ($p > 0.9$).

In contrast, the Prioritize-Search model is much more ambivalent. At $\tau = 0.5$ it classifies 14 of the 26 mixed queries as Search (53.8%) and 12 as Recommendation (46.2%), and this bias towards Search only gradually strengthens as the threshold is increased. The mean predicted probability for the priority class is markedly lower ($\overline{p}_{\text{search}} = 0.614$), with higher variance ($\sigma \approx 0.27$) and higher entropy (mean ≈ 0.48). More than half of the mixed utterances (57.7%) fall into a low-confidence region ($p < 0.6$), and only around one-third (34.6%) reach high confidence ($p > 0.9$).

Taken together, these results indicate that, in the latent space learned by DeBERTa-v3-base, mixed-intent queries are *not neutral*: they lie substantially closer to Recommendation than to pure Search. When the model is trained to prioritize Recommendation, the mixed queries are pulled almost entirely into the Recommendation region; when trained to prioritize Search, the same utterances sit on or near the decision boundary and the classifier hesitates.

Calibration and Confidence on Mixed Intents

To further characterise this behaviour, we visualise the distribution of predicted probabilities and plot calibration curves restricted to the mixed-intent set. The probability histogram for the Prioritize-Rec model shows a sharp peak near $p \approx 0.95$, corresponding to the majority of ambiguous queries being assigned extremely high Recommendation probability, and a small secondary cluster around $p \in [0.5, 0.7]$ capturing genuinely borderline cases. The corresponding calibration curve aligns closely with the ideal diagonal: bins with higher mean predicted probability also contain a higher fraction of "positive" (Recommendation-priority) instances. This suggests that the model is not arbitrarily over-confident; rather, it

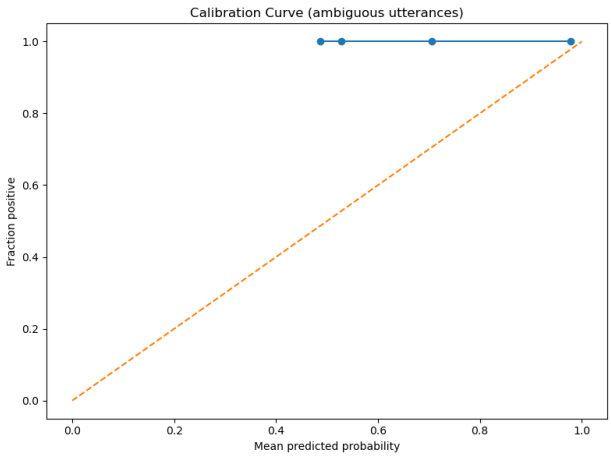


Figure 5.15: Calibration

systematically interprets almost all mixed intents as recommendation-like.

The calibration curve gives us a clearer look at the model's behavior. It shows that the Prioritize-Rec model is heavily biased where it treats almost every 'mixed' intent as a recommendation. Its confidence scores follow this pattern closely, reflecting just how frequently it correctly identifies those recommendation-heavy bins.

By contrast, the Prioritize-Search model shows a much more flatter and more dispersed distribution, where probabilities span the entire $[0.3, 0.97]$ range, and the calibration curve reveals that mid-range probability bins ($p \in [0.4, 0.7]$) contain a highly heterogeneous mix of cases. This supports the hypothesis that Search, unlike Recommendation, occupies a more diffuse and linguistically diverse region in the space of user queries.

5.6 Data Augmentation and Signal Recovery

Initial experiments under a Search-priority routing policy revealed performance diminution, including model collapse and near-zero ROC-AUC. To address this issue, a targeted data augmentation strategy was applied to strengthen the Search signal during training.

The augmentation procedure combined paraphrasing and back-translation techniques applied targetedly to Search-labeled utterances. This approach mirrors recent work in conversational search modeling, where large language models are used to synthesize linguistically diverse but semantically consistent training examples to alleviate data sparsity [27].

Following augmentation, the Search-priority model shows a substantial recovery in performance. The augmented model achieves a test F1-score of 0.899, with a precision of 0.969 and a ROC-AUC of 0.954. These results indicate that the Search and Recommendation classes become separable once sufficient linguistic variation is introduced into the training data.

From a system perspective, data augmentation transforms the Search-priority router from a malfunctioning case into a robust operational choice. The resulting model consistently routes informational queries to the open-domain corpus while minimizing incorrect personalization, permitting a stable and interpretable CSR pipeline.

5.6.1 Augmentation

The augmented models highlight the differing effects of profile information under the two policies. For Recommendation-priority, profiles improve performance; for Search-priority, they introduce bias that reduces accuracy. The results demonstrate a clear asymmetry in classifier behavior as well as the strong influence of user profile information on model performance. We used the Matthews Correlation Coefficient here, because it provides a more balanced view of binary classification than accuracy alone.

Under the Recommendation-priority policy, the model shows consistently strong performance. When using Context-Only input, the classifier achieves an accuracy of 0.761 and an F1 score of 0.839, with a high recall of 0.958, indicating that it reliably identifies Recommendation intents even in ambiguous contexts. The ROC-AUC of 0.941 confirms strong separability between Recommendation and Search signals. When the User Profile is added (Context+Profile), performance improves substantially across all metrics, reaching 0.949 accuracy and an F1 score of 0.961, the highest observed in any configuration. The ROC-AUC rises to 0.986, indicating near-perfect class discrimination. These results confirm that profile information enriches user-preference cues and further amplifies the already dominant Recommendation signal, making this policy highly robust.

In contrast, the Search-priority policy exhibits severe instability. With Context-Only, the classifier collapses toward predicting Recommendation irrespective of the intended Search label. Accuracy falls to 0.249, F1 to 0.385, and MCC becomes strongly negative (–0.577), indicating systematic misclassification rather than random noise. The ROC-AUC of 0.129 suggests that the model fails to distinguish Search-like language at all. This failure becomes even more extreme when the User Profile is included. Accuracy drops to 0.116, Search recall to 0.099, and MCC to –0.747, while ROC-AUC reaches only 0.061 a near-zero score consistent with complete collapse of the decision boundary. These findings confirm that user profile features introduce a powerful Recommendation-oriented prior, which overwhelms the Search intent signal and prevents the Search-priority model from functioning correctly.

Taken together, these results reveal a fundamental semantic imbalance in the CoSRec domain: Recommendation language dominates the latent space, while Search language remains fragile and easily overshadowed, especially in the presence of profile data. The Recommendation-priority formulations, particularly with Context+Profile perform strongly and reliably, whereas the Search-priority configurations expose the inherent difficulty of modeling open-ended, information-seeking queries. This asymmetry motivates the later augmentation interventions aimed at strengthening the Search signal.

The Role of Personalization and Bias

Under the **Recommendation-priority policy**, we assign the positive class (1) to any utterance that includes a Recommendation intent (either Rec or Rec + Search). This setup is designed to ensure the classifier catches every instance where a user might need a personalized suggestion.

A. Context-Only Input: Using only the dialogue context, the DeBERTa-v3 model performs remarkably well, achieving an F1-score of 0.927 and near-perfect Precision (0.994). This high precision is particularly interesting; it shows that the model is already very good at separating the two intents based on the conversation alone, rarely mislabeling a Search-only

query as a Recommendation.

B. Context + Profile Input: Performance improves even further when we add the user profile (F1 = 0.961, AUC = 0.986). This makes sense, as the profile provides a clear history of user preferences and past behavior—signals that are naturally tied to recommendation intents. In this case, the extra data reinforces the policy's goal of capturing as many potential recommendation scenarios as possib

In the other hand, the **Search-priority policy**, we label any utterance containing a Search intent (either pure Search or Search + Rec) as the positive class (1). Our goal here is to see how well the system can catch informational queries that need to be routed to open-domain retrieval.

C. Context-Only Input: Without any user profile data, the model struggles significantly, reaching an Accuracy of only 0.249 and an AUC of 0.129. This result highlights just how hard it is to isolate Search as the primary signal. In the CoSRec dataset, search queries are often short and highly variable; because they lack a strong, consistent structure, the model frequently mistakes them for Recommendation-like language and defaults to the negative class.

D. Context + Profile Input: Surprisingly, adding the user profile makes performance even worse, with the AUC dropping to 0.061 and an MCC of –0.747. It appears that the profile information introduces too much noise from the recommendation side, effectively drowning out the search intent. In this setup, the classifier almost always predicts the Recommendation class, leading to a near-total collapse in all performance metrics.

Chapter 6

Future work and Conclusions

6.1 Conclusions

This thesis examined the issue of intent routing within Joint Conversational Search and Recommendation (CSR), which addresses a fundamental limitation in current conversational systems: the inability to consistently differentiate between search-oriented and recommendation-oriented user needs in changing, multi-turn interactions.

To tackle this challenge, we first introduced the *CoSRec dataset*, a innovative benchmark specifically designed to model mixed-intent conversational scenarios. Unlike traditional datasets that isolate either retrieval or recommendation tasks, CoSRec enables the study of intent ambiguity by combining search, recommendation, and hybrid interactions within the same conversational framework.

Utilizing this dataset, we formulated intent routing as both a multi-label classification task and a binary decision problem under different routing policies. Initial experiments with classical approaches (e.g., TF-IDF and logistic regression) highlighted the limitations of lexical features in capturing short, context-dependent utterances. In contrast, transformer-based models, particularly DeBERTa-v3, demonstrated the best performance, using contextual embeddings to interpret sparse conversational signals.

However, a deeper analysis revealed that performance was not uniform across intents, where recommendation intents were consistently detected with high confidence, while search intents proved significantly more challenging. This led to the identification of a **semantic asymmetry**, where recommendation language is linguistically rich and explicit, whereas search queries are shorter, underspecified, and therefore harder to classify.

Through further experiments a critical failure mode was undercovered 5.2, referred to as the **Personalization Paradox**, meaning that there is a conflict between user history and the current intent. When user profile information was incorporated into the routing stage, model performance worsened significally in search-priority settings, with ROC-AUC dropping to 6.1%. This result demonstrates that personalization helps show better results, but it can make the system stop listening to what the user is actually asking for. It gets so focused on their history that it ignores their current words..

To understand better whether these limitations were due to optimization or data quality, we conducted an extensive hyperparameter sensitivity analysis. Across variations in learning

rate, batch size, and training epochs, the model shoed highly stable behavior, with F1-scores consistently plateauing around 74–75%. This stability indicates that the performance ceiling is not caused by a bad training, but rather by the naturally messy input data that make it difficult for the model to distinguish between different categories.

For this, we introduced a targeted data augmentation strategy based on LLM-generated conversational samples. This approach significantly improved the representation of under-expressed search intents, making effective signal recovery able. As a result, the search-priority routing policy improved from near-failure to robust performance, achieving up to 89.9% F1-score and 96.9% precision, while maintaining strong ROC-AUC values above 96%.

Taken together, these findings highlight that the core challenge in CSR is not model capacity, but the quality and balance of the training signal or also called the correct answers given to the model. The experiments demonstrate that even advanced architectures are highly sensitive to semantic imbalance and can fail when dominant signals overshadow the weaker ones.

If we look at a system design perspective, this thesis establishes several key principles, where first, intent routing should rely strictly on immediate context, without early integration of personalization, to avoid biasing the decision boundary. Second, data quality and representative data are essential, particularly for capturing fragile intents such as search. Third, improving performance in CSR systems requires focusing on signal clarity rather than increasing model complexity.

In summary, this work provides a structured framework for understanding and designing intent-aware conversational systems. Rather than treating search and recommendation as independent components, CSR systems must explicitly reason about intent ambiguity and dynamically adapt their behavior based on conversational context. The results presented in this thesis demonstrate that robust intent routing is achievable, but only when architectural design, data representation, and evaluation are carefully aligned.

6.2 Future Work

Synthetic benchmarks has given us a good starting point, but if we want multi-modal agents to work in the real world, we need to move toward more intuitive applications. We see three main ways to close this gap.

Refining Data Granularity and Personalization

While the *CoSRec dataset* provides a strong foundation, future work should move toward phrase-level labelling. Rather than identifying *product_ detail* intents at the utterance level, annotating the specific text segments that satisfy a request would enable more precise extractive Question Answering (QA).

Additionally, we plan to move beyond standard recommendations by embedding a user's historical context directly into the search process. This allows the system to retrieve information from large-scale datasets, such as MS-MARCO, that is specifically tuned to an individual's unique preferences and background.

Unified Architectures and Edge Optimization

Our current framework follows a work made of different stages, where intent routing is a precursor to retrieval. A major improvment would be the transition toward a unified, end-to-end neural framework. Such a model would jointly optimize the dialogue policy and the retrieval mechanism, likely resulting in gains for both latency and accuracy.

Furthermore, for these systems to be unable for daily use, the architecture must be optimized for edge deployment. Reducing the computational footprint for mobile and low-power devices is essential for transitioning CSR from a research server to a user's pocket.

Proactive and Mixed-Initiative Interaction

Finally, future work will involve expanding the system's capabilities to support more natural, human-like dialogue for future work. Future agents should not merely be reactive; they should be proactive, seeking clarification when confidence is low or asking for feedback on specific suggestions.

Transitioning toward mixed-initiative interactions will allow users to refine their requirements or shift topics mid-conversation. By supporting these dynamic pivots, conversational systems can more closely replicate the fluidity and adaptability of a human assistant.

Bibliography

[1] On the use of feature selection methods in classification. *Journal of the Royal Statistical Society*, 58(1):189–198, 1996.

[2] From data to knowledge: The dikw framework. *Journal of Information Science*, 2003.

[3] Overview of the trec conference. *Information Processing & Management*, 2005.

[4] *Search Engines: Information Retrieval in Practice*. Pearson, 2010.

[5] A survey of rule-based and statistical dialogue systems. *Computer Speech and Language*, 2010.

[6] Limitations of roc-auc in imbalanced classification. *Pattern Recognition*, 2020.

[7] Open-set intent recognition with semantic uncertainty. In *Proceedings of COLING*, 2022.

[8] Marco Alessio, Simone Merlo, Guglielmo Faggioli, Marco Ferrante, Cristina Ioana Muntean, Franco Maria Nardini, Raffaele Perego, and Giuseppe Santucci. Cosrec: A joint conversational search and recommendation dataset. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. ACM, 2025.

[9] Nicholas J Belkin, Colleen Cool, Judy Jeng, David Keller, Michael J Lynch, and Soyeon Park. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Instructional Science*, 30:65–82, 2002.

[10] Peter Brusilovsky and Daqing He, editors. *Social Information Access: Systems and Technologies*, volume 10100 of *Lecture Notes in Computer Science*. Springer, 2018.

[11] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[12] Raghu Chukkala. The evolution of generative ai in conversational systems: advancing chatbots and ccai for next-gen business intelligence. *Global Journal of Engineering and Technology Advances*, 23:195–206, 05 2025.

[13] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. Bias and unfairness in information retrieval systems: New challenges in the llm era. In

Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 1–11. ACM, 2024.

[14] A. Philip Dawid. The well-calibrated bayesian. *Journal of the Royal Statistical Society: Series B*, 44(3):605–613, 1982.

[15] Jens-Joris Decorte, Jeroen Van Hautte, Chris Develder, and Thomas Demeester. On the biased assessment of expert finding systems. In *Proceedings of the 4th Workshop on Recommender Systems for Human Resources (RecSys in HR)*, pages 1–8. CEUR-WS, 2024.

[16] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, pages 4171–4186, 2019.

[18] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[19] Nicola Ferro and Maria Maistro. Evaluation of ir systems. *ACM Computing Surveys*, 1(1):1–75, 2024.

[20] Ilyes Gaou, Farah Benamara, and Camille Pradel. Utilisation of open intent recognition models for robust dialogue systems. *Journal of Theoretical and Applied Information Technology*, 2023.

[21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*, 2021.

[22] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

[23] Vladimir et al. Karpukhin. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, 2020.

[24] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[25] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM Conference on Recommender Systems*, pages 114–122. ACM, 2017.

[26] Raymond Li, Samira Kahani, Hannes Schulz, and Layla El Asri. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[27] Xinyu Li, Honglei Zhuang, Jiashuo Wang, and W. Bruce Croft. Large language models for conversational search session synthesis. *arXiv preprint arXiv:2403.03952*, 2024.

[28] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–225, 2021.

[29] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, 2008.

[30] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[31] B.W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[33] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, et al. A survey of conversational search. *ACM Transactions on Information Systems*, 43(6):Article 167, 2025.

[34] Fengran Mo, Chuan Meng, Mohammad Aliannejadi, and Jian-Yun Nie. Conversational search: From fundamentals to frontiers in the llm era. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2025.

[35] Eli Pariser. *The Filter Bubble: How the New Personalized Web Is Changing What We Think and How We Live*. Penguin, New York, 2011.

[36] Thu Zar Phyu and Nyein Nyein Oo. Performance comparison of feature selection methods. *MATEC Web of Conferences*, 42:06002, 2016.

[37] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[38] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Introduction to Recommender Systems Handbook*. Springer, New York, NY, 2011.

[39] Stephen E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, 1977.

[40] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. NIST Special Publication, 1995.

[41] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *Nist Special Publication Sp*, volume 109, page 109. National Institute of Standards and Technology, 1995.

[42] Joseph J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[43] Gerard Salton and Christopher Buckley. *Term-weighting approaches in automatic text retrieval*, volume 24. Elsevier, 1988.

[44] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[45] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.

[46] Jaime Teevan, Susan T Dumais, and Eric Horvitz. Personalizing search: is it worth it, and how does it happen? In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 415–422. ACM, 2005.

[47] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: Modeling queries with variation in user intent. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, pages 163–170. ACM, 2008.

[48] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: Modeling queries with variation in user intent. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 163–170. ACM, 2008.

[49] Ryen W White and Robert A Roth. *Exploratory Search: Beyond the Query-Response Paradigm*. Morgan & Claypool Publishers, San Rafael, CA, 2009.

[50] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

[51] Weinan Zhang, Tianqiang Zhou, Jun Wang, and Jian Xu. Sequential selection of advertisements: Exploiting & exploring multi-armed bandits. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 734–743. ACM, 2012.

[52] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–191, 2019.

[53] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 177–186. ACM, 2018.

[54] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and William Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, pages 177–186. ACM, 2018.

[55] Kun Zhou, Xiaolei Wang, Yuanhang Ji, Pengfei Zang, Wayne Xin Zhao, and Ji-Rong Wen. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1006–1014, 2020.